



**Министерство науки и высшего образования  
Российской Федерации Федеральное государственное  
бюджетное образовательное учреждение высшего  
образования «Московский государственный  
технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)**

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

Лабораторная работа №6  
по дисциплине «Базовые компоненты интернет-технологий»

Выполнил:  
студент группы ИУ5-32Б  
Поддубный М.Н.

Проверил:  
Канев А.И.

2021 г.

## Описание задания

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

## Текст программы

main.py

```
from aiogram import types
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters import Command

from loader import dp
from states.test import Test
import random
from faker import Faker
import datetime

fake = Faker('ru_RU')

@dp.message_handler(Command("start"), state=None)
async def enter_test(message: types.Message):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    item1 = types.KeyboardButton('Кто хозяин')
    item2 = types.KeyboardButton('Случайное число от 0 до 100')
    item3 = types.KeyboardButton('Случайное ФИО')
    item4 = types.KeyboardButton('Дата и время на данный момент')
    markup.add(item1, item2, item3, item4)

    await message.answer('Привет, {0.first_name}'.format(message.from_user),
reply_markup=markup)

@dp.message_handler(Command("test"), state=None)
async def enter_test(message: types.Message):
    await message.answer("Сайт знакомств\n"
                        "АНКЕТА\n"
                        "Вопрос №1. \n\n"
                        "Ваши Фамилия и Имя?")

    await Test.Q1.set()

@dp.message_handler(state=Test.Q1)
async def answer_q1(message: types.Message, state: FSMContext):
    answer = message.text

    await state.update_data(answer1=answer)

    await message.answer("Вопрос №2. \n\n"
                        "Ваш возраст")

    await Test.Q2.set()

@dp.message_handler(state=Test.Q2)
async def answer_q1(message: types.Message, state: FSMContext):
    answer = message.text
```

```

        await state.update_data(answer2=answer)

        await message.answer("Вопрос №3. \n\n"
                               "Ваш знак зодиака")

        await Test.Q3.set()

@dp.message_handler(state=Test.Q3)
async def answer_q1(message: types.Message, state: FSMContext):
    answer = message.text

    await state.update_data(answer3=answer)

    await message.answer("Вопрос №4. \n\n"
                           "Ваш город")

    await Test.Q4.set()

@dp.message_handler(state=Test.Q4)
async def answer_q2(message: types.Message, state: FSMContext):
    answer = message.text.lower()

    await state.update_data(answer4=answer)

    await message.answer("Вопрос №5. \n\n"
                           "Ваши интересы")

    await Test.next()

@dp.message_handler(state=Test.Q5)
async def answer_q2(message: types.Message, state: FSMContext):
    # Достаем переменные
    answer = message.text
    await state.update_data(answer5=answer)
    data = await state.get_data()
    answer1 = data.get("answer1")
    answer2 = data.get("answer2")
    answer3 = data.get("answer3")
    answer4 = data.get("answer4")
    answer5 = data.get("answer5")
    await message.answer(data)
    await message.answer("УРА!!")
    await state.finish()

@dp.message_handler()
async def get_text(message: types.Message):
    if message.text == "Случайное ФИО":
        await message.answer('ФИО: ' + str(fake.name()))
    elif message.text == 'Кто хозяин':
        await message.answer('Михаил Николаевич')
    elif message.text == "Случайное число от 0 до 100":
        await message.answer('Число ' + str(random.randint(0, 100)))
    elif message.text == "Дата и время на данный момент":
        now = datetime.datetime.now()
        await message.answer('Сейчас ' + str(now.strftime("%d-%m-%Y
%H:%M:%S")))
    elif message.text.lower() == "привет":
        await message.answer("привет")
    elif message.text.lower() == 'как жизнь':

```

```

        await message.answer('потихоньку, брат, у тебя че как')
    elif message.text.lower() == 'здорова':
        await message.answer('привет брат')
    elif message.text.lower() == '1':
        await message.answer('1')
    else:
        await message.answer('хочу другое сообщение')

```

app.py

```

from loader import bot, storage

async def on_shutdown(dp):
    await bot.close()
    await storage.close()

if __name__ == '__main__':
    from aiogram import executor
    from handlers import dp

    executor.start_polling(dp, on_shutdown=on_shutdown)

```

loader.py

```

import logging

from aiogram import Bot, Dispatcher, types
from aiogram.contrib.fsm_storage.memory import MemoryStorage

import config

bot = Bot(token=config.BOT_TOKEN, parse_mode=types.ParseMode.HTML)
storage = MemoryStorage()
dp = Dispatcher(bot, storage=storage)
logging.basicConfig(format=u'%(filename)s [LINE:%(lineno)d] #%(levelname)-8s
[% (asctime)s]   %(message)s',
                    level=logging.INFO,
                    )

```

test.py

```

from aiogram.dispatcher.filters.state import StatesGroup, State

class Test(StatesGroup):
    Q1 = State()
    Q2 = State()
    Q3 = State()
    Q4 = State()
    Q5 = State()

```

Экранные формы с результатами выполнения программы

