



**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»

Отчёт по лабораторной работе №6

“ Разработка бота на основе конечного автомата для Telegram с  
использованием языка Python ”

Выполнил:

студент группы ИУ5-32Б

Милевич Артём Андреевич

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Гапанюк Ю.Е.

Подпись и дата:

Москва, 2021 г.

# Общее описание задания

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

## Текст программы

```
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()

queen_albums={
    1 : "Sheer Heart Attack",
    2 : "A Night At The Opera",
    3 : "News Of The World"
}

rhcp_albums={
    1 : "Californication",
    2 : "By The Way",
    3 : "Stadium Arcadium"
}

green_day_albums={
    1 : "Dookie",
    2 : "Insomniac",
    3 : "American Idiot"
}

queen_albums_price={
    "Sheer Heart Attack" : 7000,
    "A Night At The Opera" : 6000,
```

```

    "News Of The World" : 6000
}
rhcp_albums_price={
    "Californication" : 4200,
    "By The Way" : 3100,
    "Stadium Arcadium" : 3900
}
green_day_albums_price={
    "Dookie" : 3400,
    "Insomniac" : 3000,
    "American Idiot" : 3500
}

def summary(first_table,first_table_price,second_table,second_table_price,third_table,third_table_price,first_answer,second_answer,third_answer):
    a = first_table_price[first_table[first_answer]]
    b = second_table_price[second_table[second_answer]]
    c = third_table_price[third_table[third_answer]]
    return "Итоговая сумма заказа = " + str(a+b+c)+ "\n"

@dp.message_handler(state="*", commands=['start'])
async def starting_process(message: types.Message):
    await bot.send_message(message.from_user.id,"Привет! У тебя есть нереальная возможность купить по 1 альбому " +
        "величайших групп всех времён!\n\n1)Альбом группы Queen\n2)Альбом группы RHCP\n3)Альбом группы Green Day\n\n" +
        "Чтобы начать формировать заказ напиши /order")
    await Test.Q0.set()

@dp.message_handler(state=Test.Q1, commands=['order'])

@dp.message_handler(state=Test.Q0, commands=['order'])
async def starting_process(message: types.Message,state: FSMContext):
    await bot.send_message(message.from_user.id, "Выбери альбом группы Queen:\n1)Sheer Heart Attack - {}\n2)A Night At The Opera - {}\n3)News Of The World - {}".format(queen_albums_price[queen_albums[1]], queen_albums_price[queen_albums[2]], queen_albums_price[queen_albums[3]]))
    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,"Неправильно, попробуй ещё раз...")
    await state.update_data(q1 = answer)
    await bot.send_message(message.from_user.id,"Выбери альбом группы RHCP:\n1)Californication - {}\n2)By The Way - {}\n3)Stadium Arcadium - {}".format(rhcp_albums_price[rhcp_albums[1]], rhcp_albums_price[rhcp_albums[2]], rhcp_albums_price[rhcp_albums[3]]))
    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, "Неправильно, попробуй е
щё раз...")
    await state.update_data(q2 = answer)
    await bot.send_message(message.from_user.id, "Выбери альбом группы Green Day:\n
1)Dookie - {}\n2)Insomniac - {}\n3)American Idiot - {}".format(green_day_albums_pri
ce[green_day_albums[1]], green_day_albums_price[green_day_albums[2]], green_day_alb
ums_price[green_day_albums[3]]))
    await Test.Q3.set()

@dp.message_handler(state=Test.Q3)
async def third_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, "Неправильно, попробуй е
щё раз...")
    await state.update_data(q3 = answer)
    data = await state.get_data()
    sumcheck=summary(queen_albums,queen_albums_price,rhcp_albums,rhcp_albums_price,
green_day_albums,green_day_albums_price,data.get("q1"),data.get("q2"),data.get("q3"
))
    await bot.send_message(message.from_user.id, "Ваш заказ:\n\nАльбом Queen: {}\nАл
ьбом RHCP: {}\nАльбом Green Day: {}".format(queen_albums[data.get("q1")],rhcp_album
s[data.get("q2")],green_day_albums[data.get("q3")]))
    await bot.send_message(message.from_user.id, sumcheck)
    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)

```

## Результат выполнения программы

