



**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)**

**Базовые компоненты интернет технологий  
Отчет по лабораторной работе №6**

Студент: Макеев В. А.  
Группа: ИУ5Ц-54Б

Преподаватель: Гапанюк Ю. Е.

2021 г.

## **Лабораторная работа №6**

### **Задание**

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

## Текст программы

### 1. bot\_telegram.py

### 2. create\_bot.py

### 3. Папка: data\_base

#### 3.1 \_\_init\_\_.py

```
from data_base import sqlite_db
```

#### 3.2 sqlite\_db.py

```
import sqlite3 as sq
from create_bot import bot

def sql_start():
    global base, cur
    base = sq.connect('tanks.db')
    cur = base.cursor()
    if base:
        print('Database is connected')
        base.execute('CREATE TABLE IF NOT EXISTS menu(img TEXT, name TEXT PRIMARY
KEY, decription TEXT, price TEXT)')
        base.commit()

async def sql_add_command(state):
    async with state.proxy() as data:
        cur.execute('INSERT INTO menu VALUES (?, ?, ?, ?)', tuple(data.values()))
        base.commit()

async def sql_read(message):
    for ret in cur.execute('SELECT * FROM menu').fetchall():
        await bot.send_photo(message.from_user.id, ret[0], f'{ret[1]}\nОписание:
{ret[2]}\nКоличество выпущенных танков: {ret[-1]}')
```

### 4. Папка: handlers

#### 4.1 \_\_init\_\_.py

```
from handlers import client
from handlers import admin
```

#### 4.2 admin.py

```
from aiogram.dispatcher import FSMContext
from aiogram.dispatcher.filters.state import State, StatesGroup
from aiogram import types, Dispatcher
from aiogram.types import ReplyKeyboardRemove
from data_base import sqlite_db
```

```

from create_bot import bot
from keyboards import adm_kb

ID = None

class FSMAdmin(StatesGroup):
    photo = State()
    name = State()
    description = State()
    price = State()

#Получаем ID текущего модератора
async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Вы зашли в систему как модератор', reply_markup=adm_kb.button_case_admin)
    await message.delete()

#Начало диалога загрузки нового пункта меню
async def cm_start(message: types.Message):
    if message.from_user.id == ID:
        await FSMAdmin.photo.set()
        await message.reply('Загрузите фото', reply_markup=ReplyKeyboardRemove())

#Получаем первый ответ и пишем в словарь
async def load_photo(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['photo'] = message.photo[0].file_id
        await FSMAdmin.next()
        await message.reply("Введите название танка")

#Получаем второй ответ
async def load_name(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['name'] = message.text
        await FSMAdmin.next()
        await message.reply("Введите описание")

#Получаем третий ответ
async def load_description(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['description'] = message.text
        await FSMAdmin.next()
        await message.reply("Теперь укажите количество выпущенных танков",
reply_markup=adm_kb.button_case_admin)

#Получаем последний ответ и используем полученные данные
async def load_price(message: types.Message, state: FSMContext):

```

```

    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['price'] = float(message.text)
        await sqlite_db.sql_add_command(state)
        await state.finish()

'''#Выход из состояний
#@dp.message_handler(state="*", commands='отмена')
#@dp.message_handler(Text(equals='отмена', ignore_case=True), state="*")
async def cancel_handler(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        current_state = await state.get_state()
        if current_state is None:
            return
        await state.finish()
        await message.reply('OK')'''
#Регистрируем хендлеры

def register_handlers_admin(dp: Dispatcher):
    dp.register_message_handler(cm_start, commands=['Загрузить'], state=None)
    dp.register_message_handler(load_photo, content_types=['photo'],
state=FSMAdmin.photo)
    dp.register_message_handler(load_name, state=FSMAdmin.name)
    dp.register_message_handler(load_description, state=FSMAdmin.description)
    dp.register_message_handler(load_price, state=FSMAdmin.price)
    dp.register_message_handler(make_changes_command, commands='moderator',
is_chat_admin=True)

```

### 4.3 client.py

```

from aiogram import types, Dispatcher
from create_bot import bot
from keyboards import kb_client
from aiogram.types import ReplyKeyboardRemove
from data_base import sqlite_db

async def command_start(message: types.Message):
    try:
        await bot.send_message(message.from_user.id, 'Здравствуйте! Вас
приветствует бронетанковый музей-бот', reply_markup=kb_client)
    except:
        await message.reply('Общение с ботом через ЛС, напишите ему /ссылка на
бота/')

async def tour_open_command(message: types.Message):
    await bot.send_message(message.from_user.id, 'Пн-пт 9:00-20:00')

async def tour_place_command(message: types.Message):
    await bot.send_message(message.from_user.id, 'м.Бауманская')

```

```

async def tour_remove_command(message: types.Message):
    await bot.send_message(message.from_user.id, 'Успешно',
reply_markup=ReplyKeyboardRemove())

async def tour_menu_command(message: types.Message):
    await sqlite_db.sql_read(message)

def register_handlers_client(dp: Dispatcher):
    dp.register_message_handler(command_start, commands=['start', 'help'])
    dp.register_message_handler(tour_open_command, commands=['Режим_работы'])
    dp.register_message_handler(tour_place_command, commands=['Расположение'])
    dp.register_message_handler(tour_remove_command, commands=['Завершить'])
    dp.register_message_handler(tour_menu_command,
commands=['Посмотреть_список_танков'])

```

## 5. Папка: keyboards

### 5.1 \_\_init\_\_.py

```

from keyboards.client_kb import kb_client

```

### 5.2 adm\_cd.py

```

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

```

#Кнопки клавиатуры админа

```

button_load = KeyboardButton('/Загрузить')

```

```

'''button_delete = KeyboardButton('/Удалить')'''

```

```

button_case_admin = ReplyKeyboardMarkup(resize_keyboard=True).add(button_load)

```

### 5.3 client\_cd.py

```

from aiogram.types import ReplyKeyboardMarkup, KeyboardButton #
,ReplyKeyboardRemove

```

```

b1 = KeyboardButton('/Режим_работы')

```

```

b2 = KeyboardButton('/Расположение')

```

```

b3 = KeyboardButton('/Посмотреть_список_танков')

```

```

b4 = KeyboardButton('/Завершить')

```

```

kb_client = ReplyKeyboardMarkup(resize_keyboard=True)







```

```

kb_client.row(b1, b2).row(b3, b4)

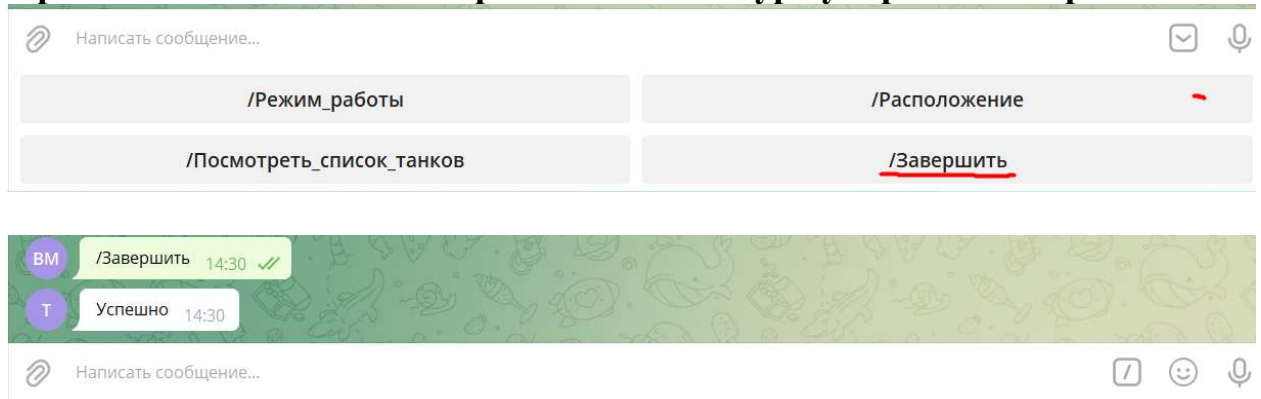
```

## 6. База данных: tanks.db

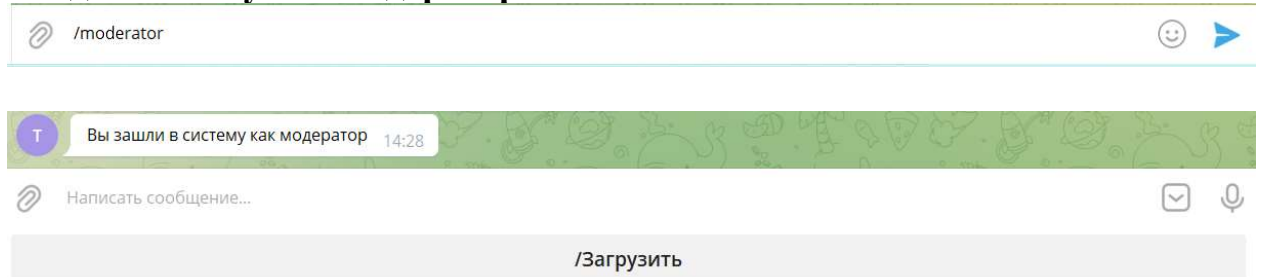
Структура БД    Данные    Прагмы    SQL				
Таблица: <input type="text" value="menu"/>       Filter in any column				
	img	name	decription	price
	Фильтр	Фил...	Фильтр	Фильтр
1	AgACAgIAAxkBAAMYcRZZ_46p9Tp...	T-34	T-34 — советский средний танк ...	35000.0
2	AgACAgIAAxkBAAMvYcRZv9SWUobit...	T-54	T-54 (Индекс ГБТУ — Объект 137) ...	16000.0

## Тест программы

При нажатии кнопки «Завершить» клавиатура убирается с экрана:



Вход в систему как модератор:



Загрузка новых данных:

Т

Вы зашли в систему как модератор 14:11

ВМ

/Загрузить 14:11 ✓✓

Виталий Макеев

/Загрузить

Т

Загрузите фото 14:11



ВМ



Виталий Макеев

Фотография

Т

Введите название танка 14:11

ВМ

Т-34 14:11 ✓✓





### Вывод новых данных:





#### T-54

Описание: Т-54 (Индекс ГБТУ — Объект 137) — советский средний танк[3]. Принят на вооружение Советской Армии ВС СССР в 1946 году, серийно выпускался с 1947 года, постоянно модернизируясь.

Количество выпущенных танков: 16000.0

T

14:29