|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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

* 1. **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]}')

1. **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'])

1. **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')

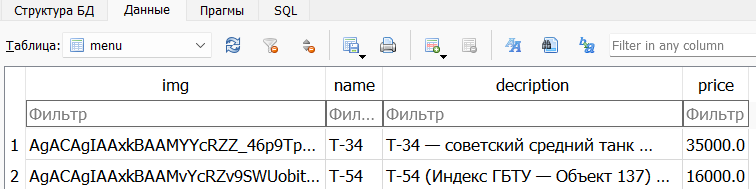
b3 = KeyboardButton('/Посмотреть\_список\_танков')

b4 = KeyboardButton('/Завершить')

kb\_client = ReplyKeyboardMarkup(resize\_keyboard=True)

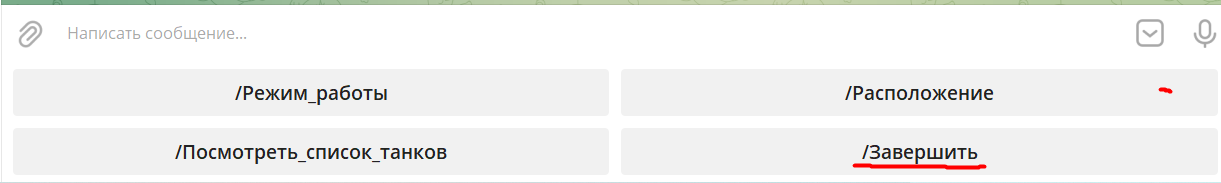
kb\_client.row(b1, b2).row(b3, b4)

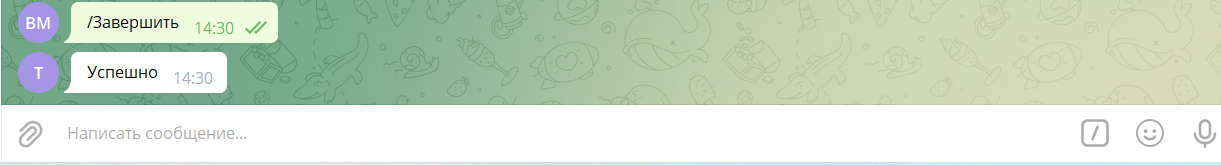
1. **База даных: tanks.db**



**Тест программы**

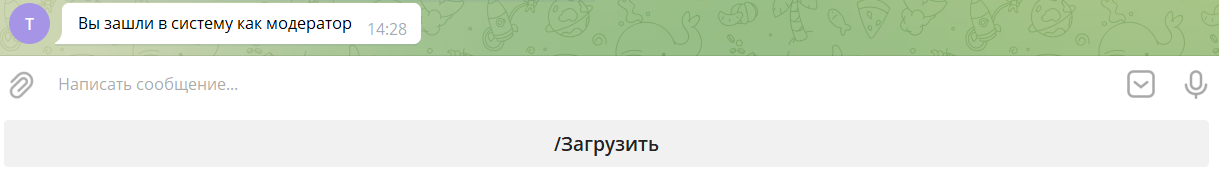
**При нажатии кнопки «Завершить» клавиатура убирается с экрана:**



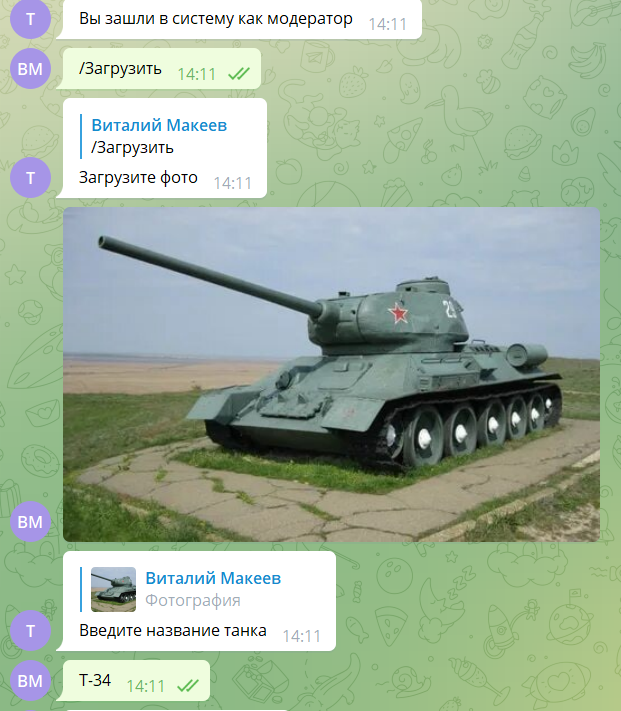


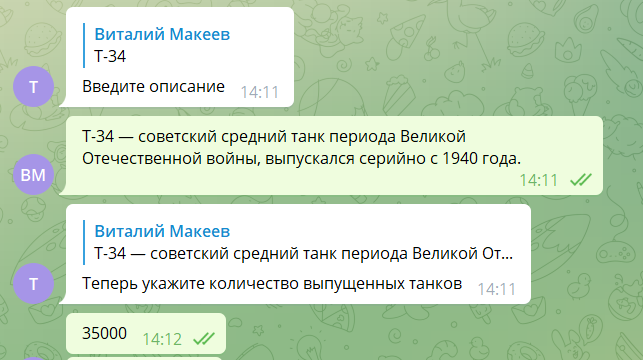
**Вход в систему как модератор:**





**Загрузка новых данных:**





**Вывод новых данных:**



