



**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 «ГУИМЦ»**

**Кафедра ИУ5 «Системы обработки информации и управления»**

Дисциплина «Базовые компоненты ИТ»

ОТЧЕТ

Лабораторная работа №5,6

Студент: Федюнев А. Ю., группа ИУ5Ц-52Б

Преподаватель: Гапанюк Ю. Е.

2021г.

## 1. Описание задания

1) Разработайте простого бота для Telegram. Бот должен использовать функциональность создания кнопок. 2) Вместо реализации паттерна Вы можете написать тесты для своей программы решения биквадратного уравнения. В этом случае, возможно, Вам потребуется доработать программу решения биквадратного уравнения, чтобы она была пригодна для модульного тестирования.

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

## 2. Текст программы

### bot\_telegram.py

```
from aiogram.utils import executor
from create_bot import dp
from data_base import sqlite_db

async def on_startup(_):
    print('Бот вышел в онлайн')
    sqlite_db.sql_start()

from handlers import client, admin

client.register_handlers_client(dp)
admin.register_handlers_admin(dp)

executor.start_polling(dp, skip_updates = True, on_startup=on_startup)
```

### create\_bot.py

```
from aiogram import Bot
from aiogram.dispatcher import Dispatcher
import os
from aiogram.contrib.fsm_storage.memory import MemoryStorage

storage=MemoryStorage()

bot = Bot(token=os.getenv('TOKEN'))
dp = Dispatcher(bot, storage=storage)
```

### keyboards/\_\_init\_\_.py

```
from keyboards.client_kb import kb_client
```

### keyboards/admin\_kb.py

```
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

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

button_case_admin =
```

```
ReplyKeyboardMarkup(resize_keyboard=True).add(button_load)\
    .add(button_delete)
```

## keyboards/client\_kb.py

```
from aiogram.types import ReplyKeyboardMarkup, KeyboardButton

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

button_case_admin = ReplyKeyboardMarkup(resize_keyboard=True).add(button_load)\
    .add(button_delete)
```

## handlers/\_\_init\_\_.py

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

## handlers/admin.py

```
from aiogram.dispatcher.filters import Text
from data_base import sqlite_db
from keyboards import admin_kb

ID = None

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

async def make_changes_command(message: types.Message):
    global ID
    ID = message.from_user.id
    await bot.send_message(message.from_user.id, 'Новые картинки?', reply_markup=admin_kb.button_case_admin)
    await message.delete()

@dp.message_handler(commands='Загрузить', state=None) #начало диалога загрузки пункта меню
async def cm_start(message : types.Message):
    if message.from_user.id == ID:
        await FSMAdmin.photo.set()
        await message.reply('Загрузить фото')

@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')

@dp.message_handler(content_types=['photo'], state=FSMAdmin.photo) #Ловим 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("Название")

#@dp.message_handler(state=FSMAdmin.name) #ЛОВИМ 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("Описание")

#@dp.message_handler(state=FSMAdmin.description)
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("Автор")

#@dp.message_handler(state=FSMAdmin.price)
async def load_author(message: types.Message, state: FSMContext):
    if message.from_user.id == ID:
        async with state.proxy() as data:
            data['author'] = message.text

        await sqlite_db.sql_add_command(state)
        await state.finish()

def register_handlers_admin(dp: Dispatcher):
    dp.register_message_handler(cm_start, commands=['Загрузить'], state=None)
    dp.register_message_handler(cansel_handler, state="*", commands='отмена')
    dp.register_message_handler(cansel_handler, Text(equals = 'отмена', ignore_case=True), state="*")
    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_author, state=FSMAdmin.author)
    dp.register_message_handler(make_changes_command, commands=['moderator'], is_chat_admin=True)

```

## handlers/client.py

```

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

#@dp.message_handler(commands=['start', 'help'])
async def commands_start(message: types.Message):
    try:
        await bot.send_message(message.from_user.id, 'Welcome', reply_markup=kb_client)
        await message.delete()
    except:
        await message.reply('Общение с ботом через личные сообщения, напишите ему:\nBKIT_DZ_telegram_bot')

```

```

#@dp.message_handler(commands=['Понедельник'])
async def pon_open_command(message: types.Message):
    await bot.send_message(message.from_user.id, 'День тяжелый... Работаем!')

#@dp.message_handler(commands=['Темы'])
async def theme_open_command(message: types.Message):
    await bot.send_message(message.from_user.id, 'ховерборд')

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

def register_handlers_client(dp : Dispatcher):
    dp.register_message_handler(commands_start, commands=['start', 'help'])
    dp.register_message_handler(pon_open_command, commands=['Понедельник'])
    dp.register_message_handler(theme_open_command, commands=['Темы'])
    dp.register_message_handler(pict_menu_command, commands=['Галерея'])

```

### data\_base/sqlite\_db.py

```

import sqlite3 as sq
from create_bot import dp, bot

def sql_start():
    global base, cur
    base = sq.connect('puct.db')
    cur = base.cursor()
    if base:
        print('Data base connected OK!')
        base.execute('CREATE TABLE IF NOT EXISTS menu(img TEXT, name TEXT PRIMARY
KEY, description 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.val-
ues()))
        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]}\nАвтор {ret[-1]}')

```

## 3. База данных

DB Browser for SQLite - D:\pystdy\telegram\_bot\puct.db

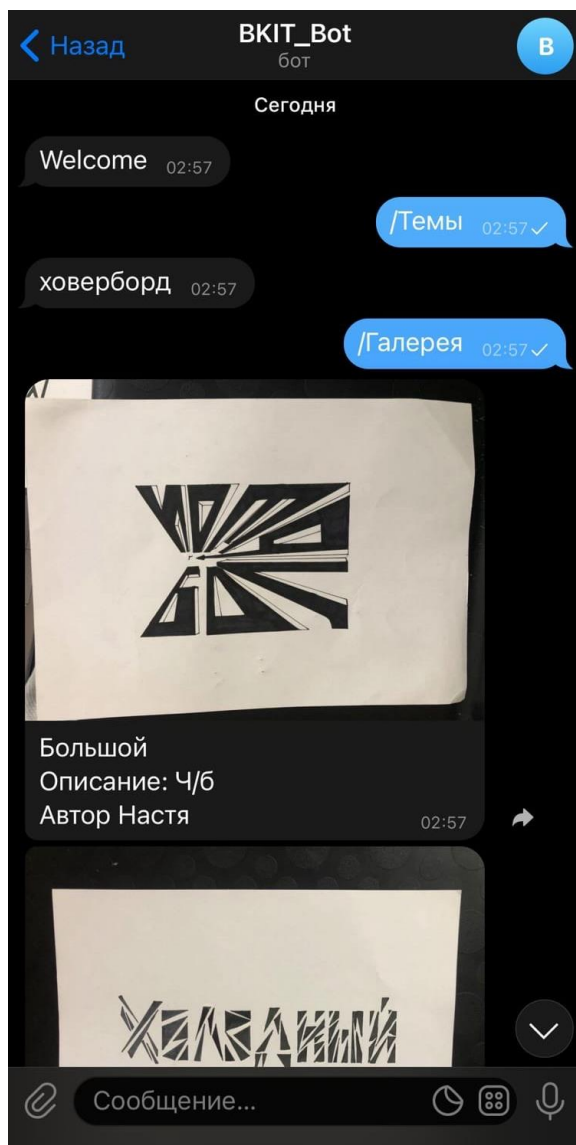
Файл Редактирование Вид Инструменты Справка

Новая база данных Открыть базу данных Записать изменения Отменить изме

Структура БД Данные Прагмы SQL

Таблица: menu

	img	name	description	price
	Фильтр	Фильтр	Фильтр	Фи...
1	AgACAgIAAxkBAANjYcpNI1Gn_moQ0o8gWmUi7T...	Большой	Ч/б	Настя
2	AgACAgIAAxkBAANsYcpNp7u9Y_hvAYmkIwu6rQJ...	Холодный	Ч/б	Настя
3	AgACAgIAAxkBAAN1YcpN514yPSuzmUeOPSi3riS...	Ярко	Яркая картинка	Саша



[Назад](#)

BKIT\_Bot

бот

В

Автор Настя

02:57

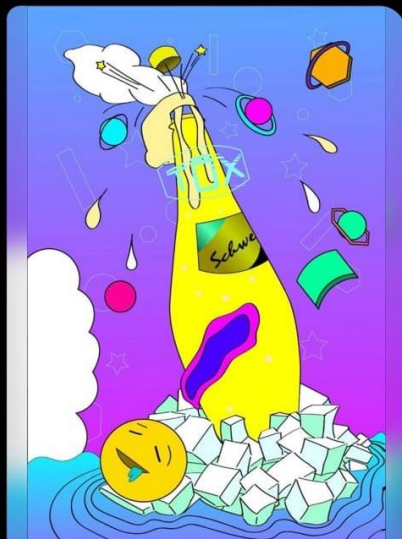


Холодный

Описание: Ч/б

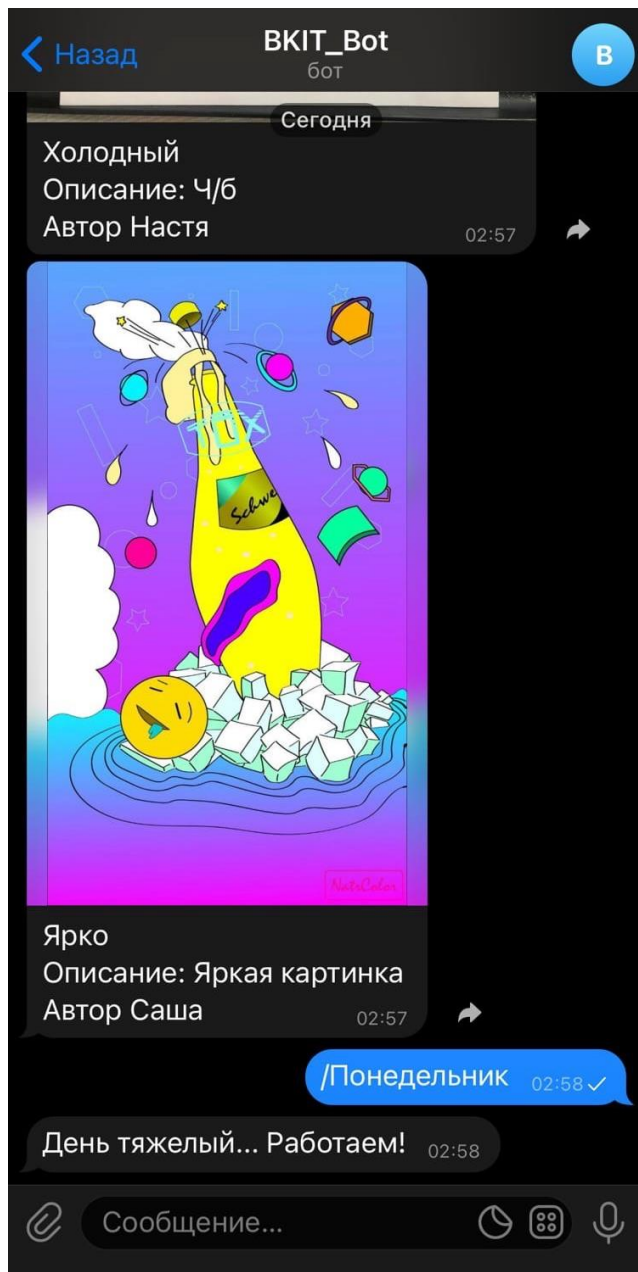
Автор Настя

02:57

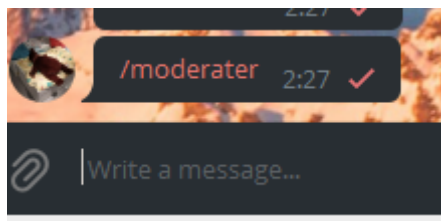


Сообщение...



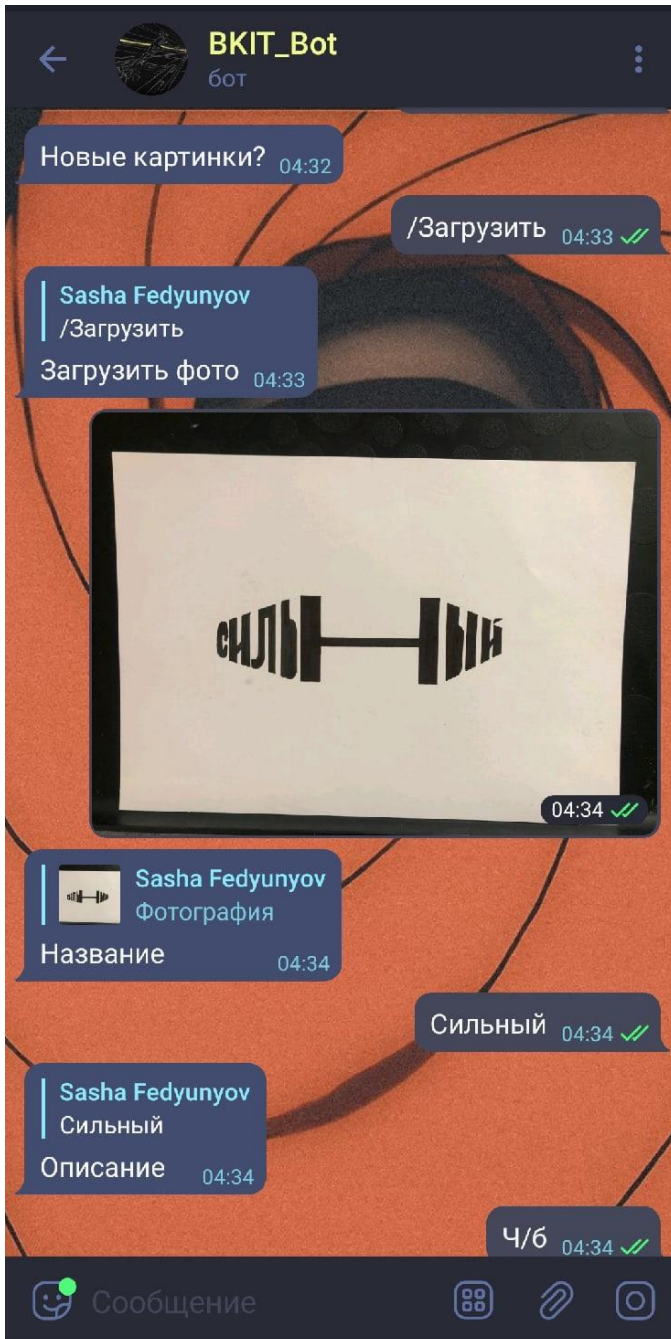
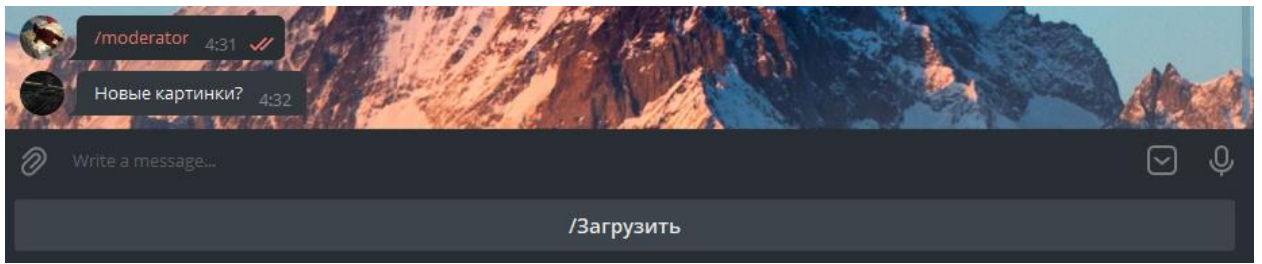


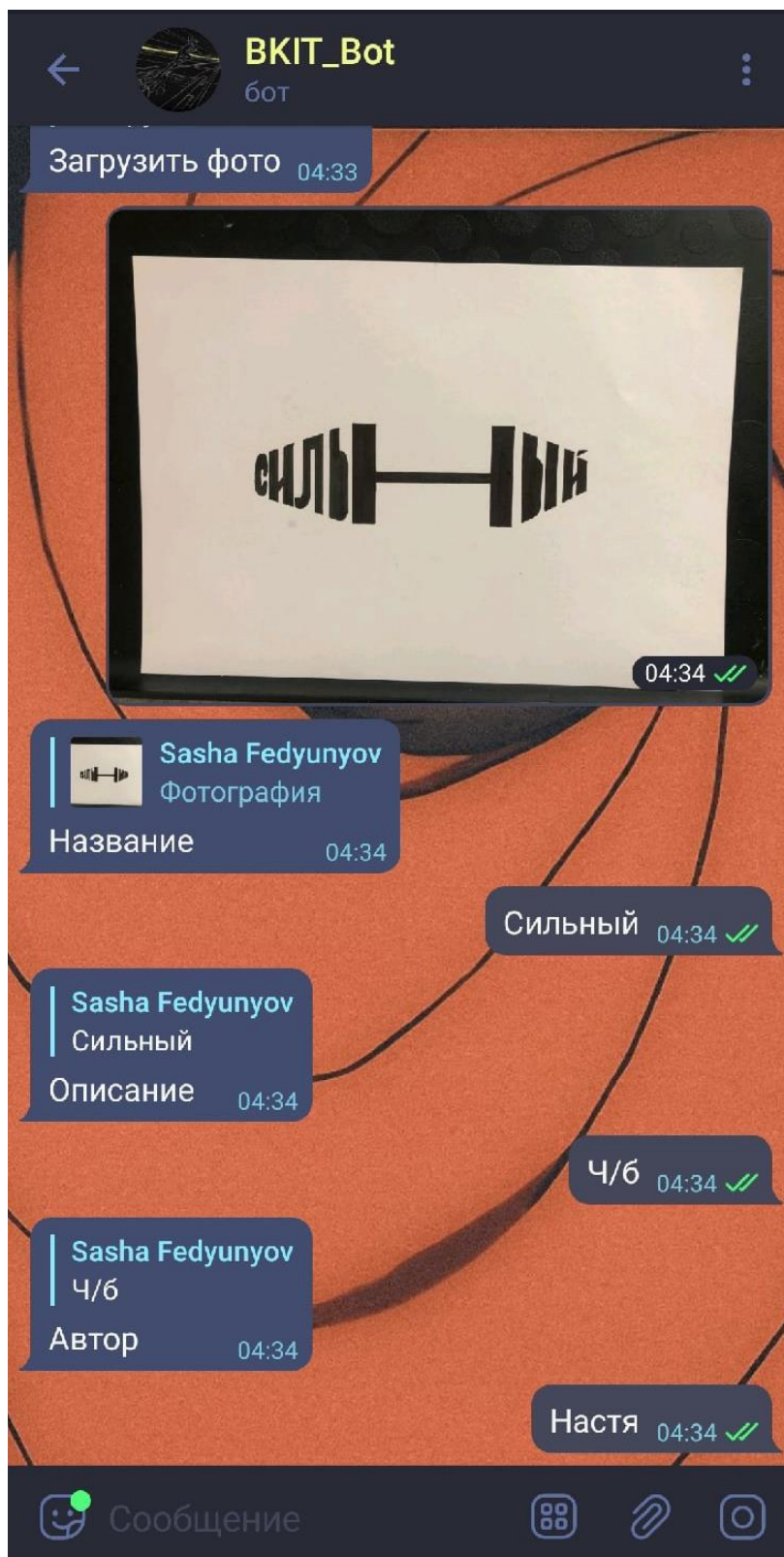
**Вход в систему как модератор:**



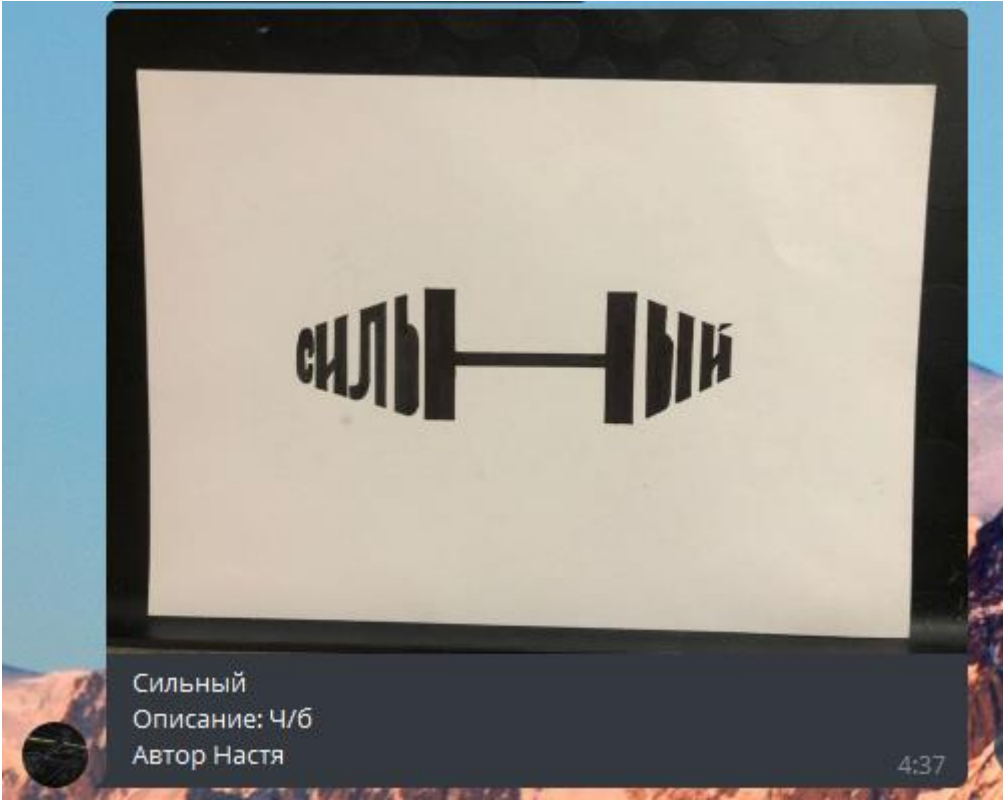
**Загрузка новых картинок**







**Вывод новых данных:**



Сильный

Описание: 4/6  
Автор Настя

4:37