

Защищено:
Папин А.В..

Демонстрация:
Папин А.В..

"__" _____ 2022 г.

"__" _____ 2022 г.

**Отчет по лабораторной работе №6 по курсу
базовые компоненты интернет-технологий (БКИТ)**

**Тема работы: "Разработка бота на основе конечного автомата для
Telegram с использованием языка Python."**

14

(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5Ц-54Б
Папин Алексей

Гапанюк Ю.Е.

(подпись)

"__" _____ 2022 г.

СОДЕРЖАНИЕ ОТЧЕТА

1. Цель лабораторной работы.....	2
2. Описание задания.	2
3. Листинг программы:	3
3.1. config.py	3
3.2. calculate_arifmetic.py	3
4.1. calculate_bot.py.....	4
4.2. json_function.py	7
4.3. work_with_calculate.py	7
4.4. bmstu.jpg	8
5. Результаты работы программы в Telegram.....	9
5.1. Получение справочную информацию	9
5.2. Основное меню переключателя	9
5.3. Простейший калькулятор (при нажатии на кнопку «Посчитать»)	9
5.4. Данные хранятся в БД в формате JSON.....	10
5.5. После несколько вычислений	10
5.6. Обновленная БД	11
5.7. Чтение и просмотр БД в Телеграме (после нажатии на кнопку Посмотреть историю вычисления)	12
5.8. Просмотр фотографии (после нажатии на кнопку «Показать фото МГТУ им. Н.Э. Баумана»)	13

1. Цель лабораторной работы

Изучение разработки ботов в Telegram.

2. Описание задания.

Выберите любой фрагмент кода из лабораторных работ 1 или 2 или 3-4.

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

3. Листинг программы:

3.1.config.py

```
token = ''
```

3.2.calculate_arifmetic.py

```
4.
# Преобразование строкового типа в list
def delete_space_into_list(string):
    new_str = []
    str_value = ''
    for i in string:
        if(i != ' '):
            str_value += i
        else:
            new_str.append(str_value)
            str_value = ''

    new_str.append(str_value)

    return new_str

# Расстановка приоритета операции
def enumeration_sign(list_str):
    count_list = []
    for i in list_str:
        if ('*' == i): count_list.append(i)
        if ('/' == i): count_list.append(i)
        if ('+' == i): count_list.append(i)
        if ('-' == i): count_list.append(i)

    count_list = prioritет(count_list)

    return count_list

# Поддержка функции по расстановку приоритета операции
def prioritет(list_str):
    new_list = []
    size = len(list_str)
    count = 0
    while (size != 0):
        if('*' in list_str or '/' in list_str):
            for i in list_str:
                if(i == '*' or i == '/'):
                    new_list.append(i)
            size -= 1
        if('+ ' in list_str or '- ' in list_str):
            for i in list_str:
                if(i == '+' or i == '-'):
                    new_list.append(i)
            size -= 1
```

```

        return new_list

# Арифметические операции
def arifmetic(sign, list):
    result = None
    if (sign in list):
        for i in range(1, len(list) - 1):
            try:
                if(list[i] == sign):
                    if(sign == '*'): result = float(list[i - 1]) *
float(list[i + 1])
                    elif(sign == '/'): result = float(list[i - 1]) /
float(list[i + 1])
                    elif (sign == '+'): result = float(list[i - 1]) +
float(list[i + 1])
                    elif (sign == '-'): result = float(list[i - 1]) -
float(list[i + 1])

                    list[i] = result
                    del list[i - 1: i]
                    del list[i: i + 1]
            except:
                return result

def calculate(value):
    new_list = delete_space_into_list(value)
    list_en = enumeration_sign(new_list)

    for sgin in list_en:
        arifmetic(sgin, new_list)

    print(float(new_list[0]))
    return float(new_list[0])

```

4.1.calculate_bot.py

```

import config
import telebot
from telebot import types
import random

from calculate_arifmetic import calculate
from calculate.work_with_calculate import get_info
from json_function import merge_data

# Создание бота
bot = telebot.TeleBot(config.token)

HELP = '''
/start - Меню переключателя
/calculate - Калькуляторный бот, способный вычислять простейшие
арифметические операции
/get_info - Просмотр историю вычисления с БД
/photo - Просмотр фото МГТУ им. Н.Э. Баумана

```

```

'''

# Справочник
@bot.message_handler(commands=['help'])
def start(message):
    bot.send_message(message.chat.id, HELP)

@bot.message_handler(commands=['start'])
def start(message):
    markup = types.InlineKeyboardMarkup(row_width=1)
    btn1 = types.InlineKeyboardButton(text="Посчитать", callback_data='btn1')
    btn2 = types.InlineKeyboardButton(text="Посмотреть историю вычисления",
callback_data='btn2')
    btn3 = types.InlineKeyboardButton(text="Показать фото МГТУ им. Н.Э.
Баумана", callback_data='btn3')
    markup.add(btn1, btn2, btn3)
    bot.send_message(message.chat.id,
                    text=f"Привет, {message.from_user.first_name}! Я
тестовый бот, выберите действия",
                    reply_markup=markup)

# Функция переключателя
@bot.callback_query_handler(func=lambda callback: callback.data)
def check_callback_data(callback):
    if (callback.data == "btn1"):
        bot.send_message(callback.message.chat.id, 'Калькулятор бот')
        bot.send_message(callback.message.chat.id, 'Напишите в чате
вычисления')

    @bot.message_handler(content_types=["text"])
    def echo(message):
        # Пользовательский идентификатор
        user_id = str(message.from_user.id)

        value = calculate(message.text)
        bot.send_message(message.chat.id, f'Решение: {value}')
        data = {
            user_id: [
                {"id": random.randint(0, 10000),
                 "value": str(message.text),
                 "result": str(value)}
            ]
        }
        merge_data(data, str(message.from_user.id))

    elif(callback.data == "btn2"):
        bot.send_message(callback.message.chat.id, 'История вычисления')
        data = get_info()
        for i in data:
            for j in data[i]:
                id = j['id']
                value = j['value']

```

```

        result = j['result']
        print_info = f'id: {id}\n{value} = {result}\n\n'
        bot.send_message(callback.message.chat.id, print_info)

    elif(callback.data == "btn3"):
        img = open('bmstu.jpg', 'rb')
        bot.send_photo(callback.message.chat.id, img)
    else:
        bot.send_message(callback.chat.id, 'Нет такой команды. Введите /help')

# Вычисления
@bot.message_handler(commands=['calculate'])
def start_calculate(message):
    bot.send_message(message.chat.id, 'Калькулятор бот')
    bot.send_message(message.chat.id, 'Напишите в чате вычисления')

    # Пользовательский идентификатор
    user_id = str(message.from_user.id)

    @bot.message_handler(content_types=["text"])
    def echo(message):
        value = calculate(message.text)
        bot.send_message(message.chat.id, f'Решение: {value}')
        data = {
            user_id: [
                {"id": random.randint(0, 10000),
                 "value": str(message.text),
                 "result": str(value)}
            ]
        }
        merge_data(data, str(message.from_user.id))

# Просмотри история вычисления
@bot.message_handler(commands=['get_info'])
def start_get_info(message):
    bot.send_message(message.chat.id, 'История вычисления')

    data = get_info()
    if (data == 'Файл отсутствует'):
        bot.send_message(message.chat.id, 'База данных отсутствует')
    else:
        for i in data:
            for j in data[i]:
                id = j['id']
                value = j['value']
                result = j['result']
                print_info = f'id: {id}\n{value} = {result}\n\n'
                bot.send_message(message.chat.id, print_info)

@bot.message_handler(commands=['photo'])
def url(message):
    img = open('bmstu.jpg', 'rb')

```

```

bot.send_photo(message.chat.id, img)

bot.polling(none_stop=True)

```

4.2.json_function.py

```

import json

def write_data(data, title='D:\Python\BKIT\calculate\data'):
    with open(f"{title}.json", "w", encoding="utf-8") as file:
        json.dump(data, file, indent=2, ensure_ascii=False)

def load_data(title="D:\Python\BKIT\calculate\data"):
    with open(f"{title}.json", "r") as file:
        data = json.load(file)
    return data

def merge_data(data_json, id_user='id_user',
title="D:\Python\BKIT\calculate\data"):
    # Если файл существует и не пустой
    try:
        with open(f"{title}.json", encoding="utf-8") as file:
            data = json.load(file)
            temp = data[id_user]
            for info_data in data_json[id_user]:
                y = {
                    'id': info_data['id'],
                    'value': info_data['value'],
                    'result': info_data['result']
                }
                temp.append(y)
            write_data(data)
    # Если файл не существует
    except:
        write_data(data_json)

```

4.3.work with calculate.py

```

import random

from calculate.json_function import write_data, load_data, merge_data
from calculate.calculate_arifmetic import calculate

def generate_value(id_user='id_user'):
    arifmetic = ['+', '-', '/', '*']

    af = arifmetic[random.randint(0, 3)]

```



```

gen_id = random.randint(0, 1000000)
v1 = random.randint(0, 1000)
v2 = random.randint(0, 1000)
result = calculate(str(v1) + ' ' + str(v2))

data = {
    str(id_user): [
        {
            "id": gen_id,
            "value": v1,
            "result": result
        }
    ]
}

merge_data(data, id_user)

def get_info():
    try:
        data = load_data()
        return data
    except:
        return 'Файл отсутствует'

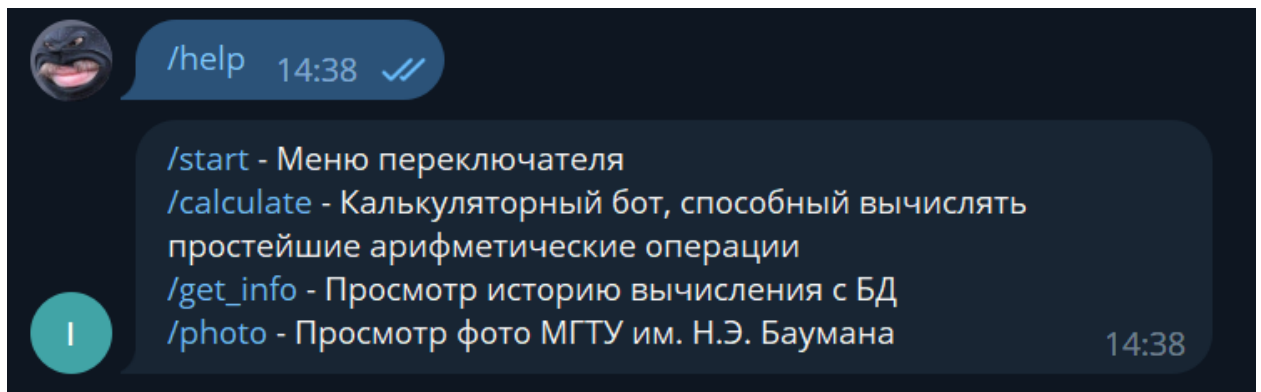
```

4.4.bmstu.jpg

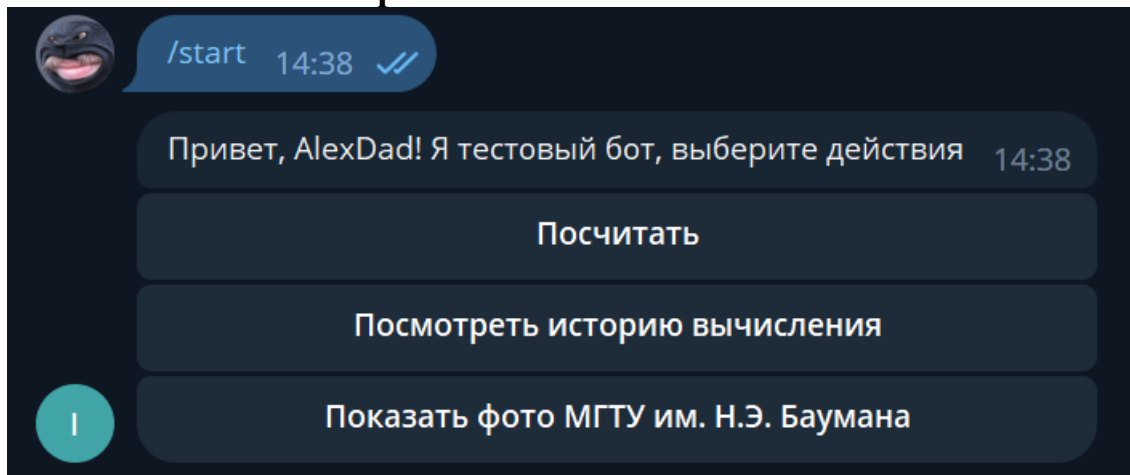


5. Результаты работы программы в Telegram

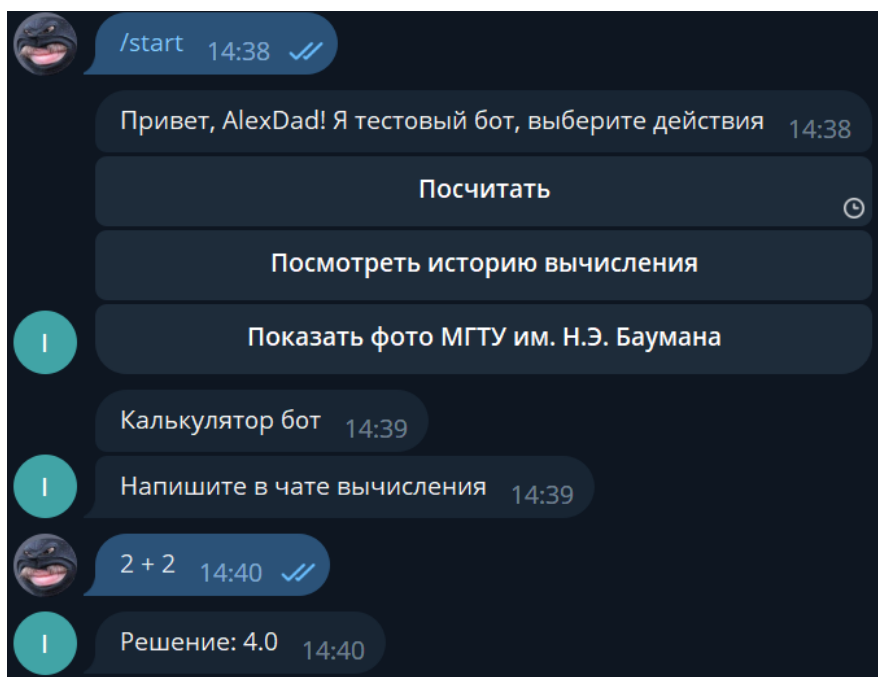
5.1. Получение справочную информацию




5.2. Основное меню переключателя



5.3. Простейший калькулятор (при нажатии на кнопку «Посчитать»)

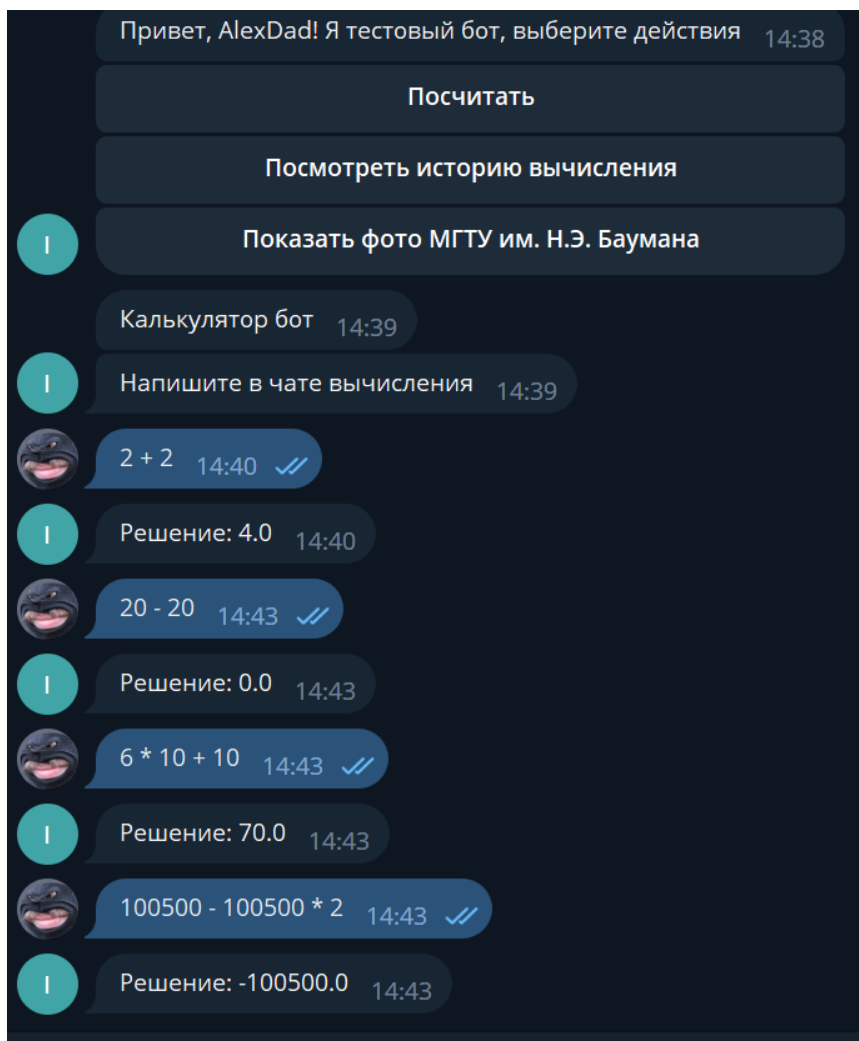


5.4. Данные хранятся в БД в формате JSON

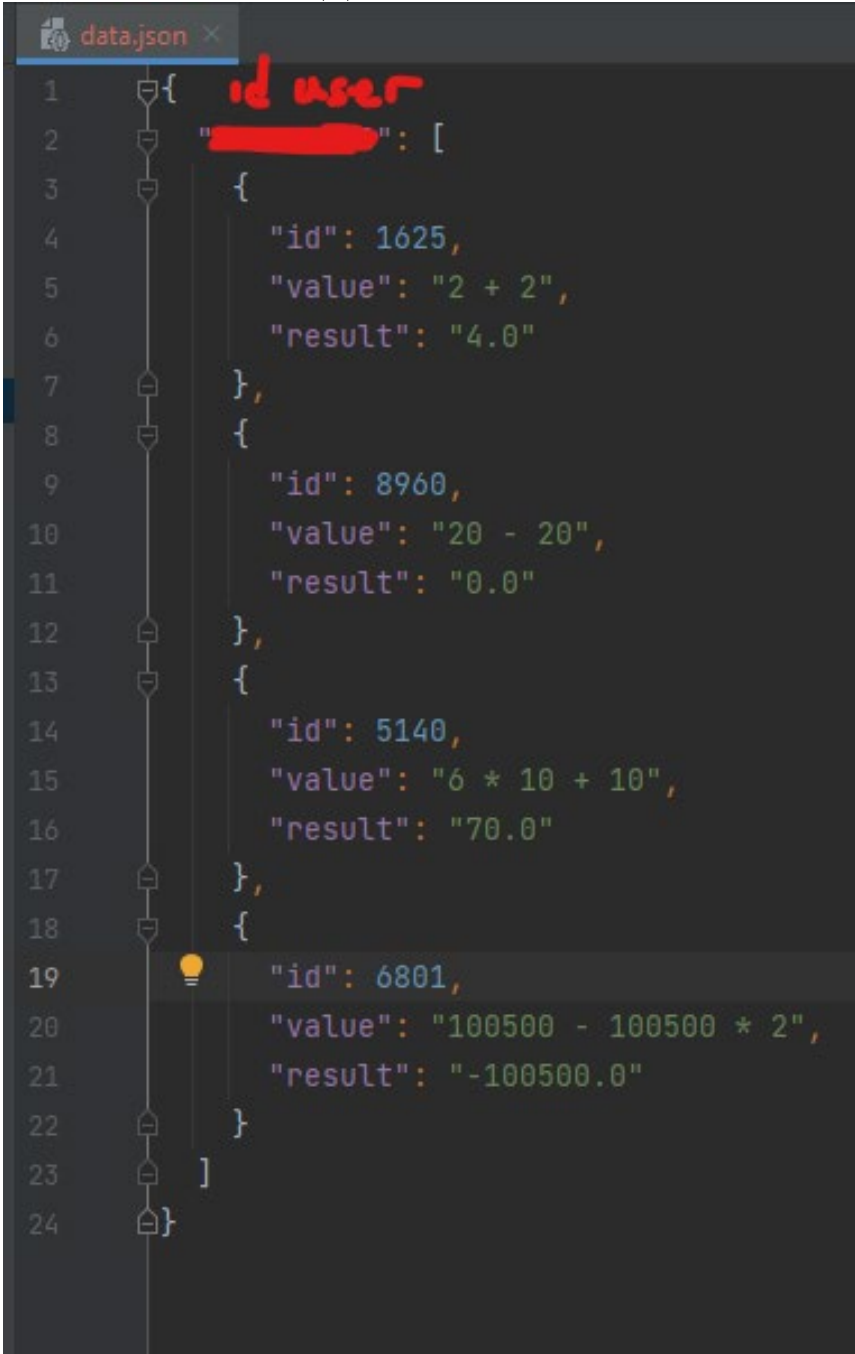


```
1 {  
2   "id user": [  
3     {  
4       "id": 1625,  
5       "value": "2 + 2",  
6       "result": "4.0"  
7     }  
8   ]  
9 }
```

5.5. После несколько вычислений

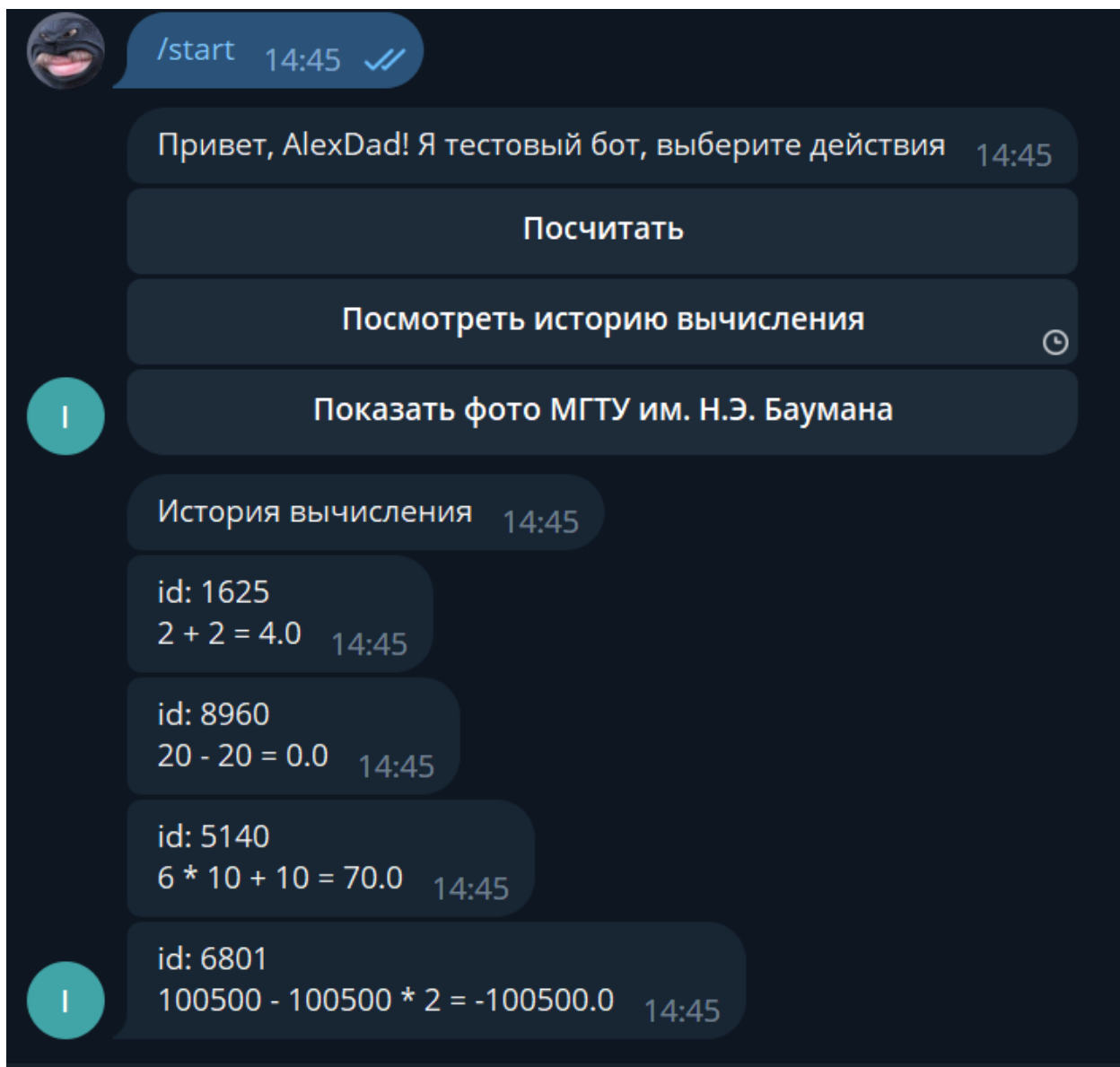


5.6.Обновленная БД



```
1  {  
2    "id user": [  
3      {  
4        "id": 1625,  
5        "value": "2 + 2",  
6        "result": "4.0"  
7      },  
8      {  
9        "id": 8960,  
10       "value": "20 - 20",  
11       "result": "0.0"  
12     },  
13     {  
14       "id": 5140,  
15       "value": "6 * 10 + 10",  
16       "result": "70.0"  
17     },  
18     {  
19       "id": 6801,  
20       "value": "100500 - 100500 * 2",  
21       "result": "-100500.0"  
22     }  
23   ]  
24 }
```

5.7. Чтение и просмотр БД в Телеграме (после нажатии на кнопку Посмотреть историю вычисления)



5.8. Просмотр фотографии (после нажатии на кнопку «Показать фото МГТУ им. Н.Э. Баумана»)

