

Чат-бот – Fасеру

Автор – Дьячкова Дарья.

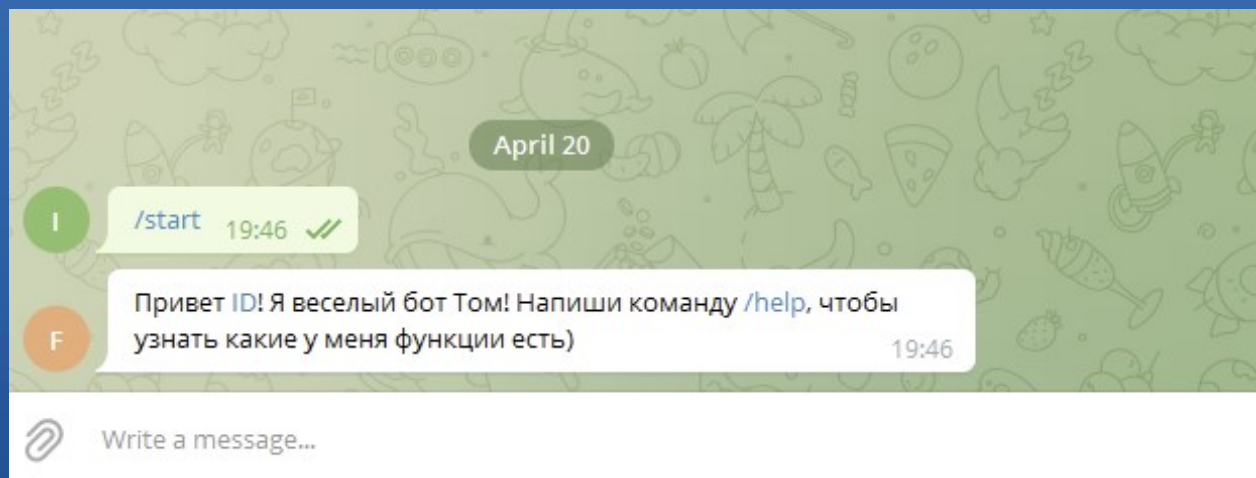
О проекте

Мой проект представляет собой чат-бот под названием Фасеру. В этом развлекательном приложении можно поэкспериментировать с фотографиями людей и конечно же поиграть!)

Функции моего бота

- /start – приветствует пользователя.
- /help – объясняет пользователю как пользоваться данным ботом.
- /quiz – включает пользователю игру, в которой надо будет угадать, что за знаменитая личность изображена на ней.
- /search – если пользователь вводит имя и фамилию известной личности, то бот выводит его фотографию.
- /guess – присылаешь моему боту фотографию человека, а он (бот) пишет кто это.
- /makeup – делает макияж.

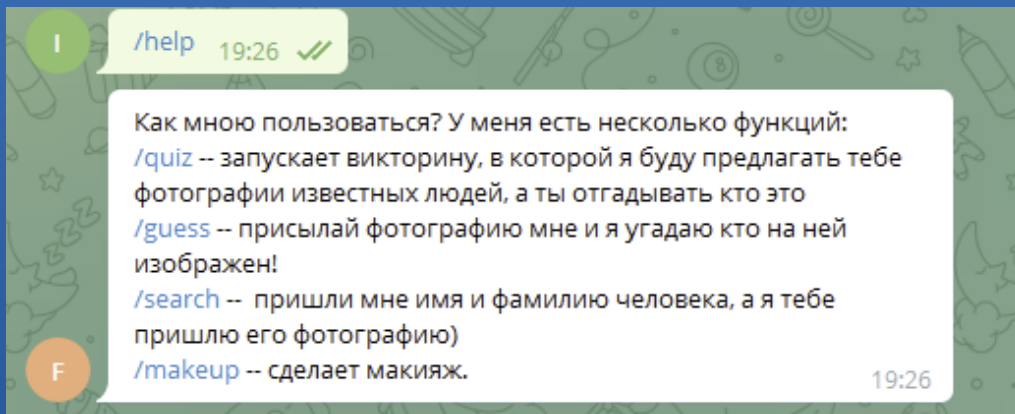
Функция start



Код для функции start

```
async def start(update, context):  
    global parametr  
    """Отправляет сообщение когда получена команда /start"""  
    user = update.effective_user  
    await update.message.reply_html(  
        rf"Привет {user.mention_html()}! Я веселый бот Том! Напиши команду /help, чтобы узнать какие у меня функции есть)")
```

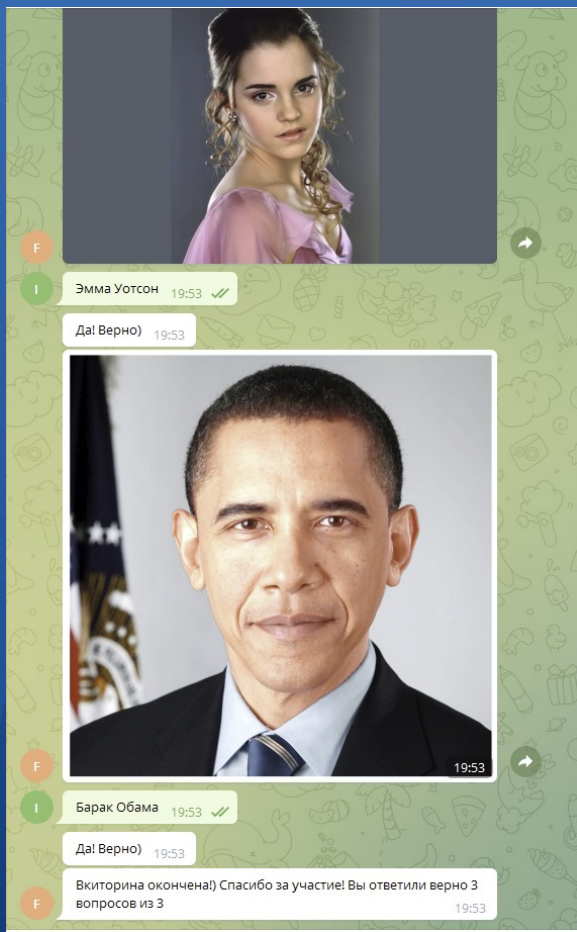
Функция help



Код для функции help

```
async def help(update, context):
    """Отправляет сообщение когда получена команда /help"""
    await update.message.reply_text("Как мною пользоваться? "
                                     "У меня есть несколько функций: \n"
                                     "/quiz -- запускает викторину, в которой я буду предлагать тебе фотографии известных "
                                     "людей, а ты отгадывать кто это \n"
                                     "/guess -- присылай фотографию мне и я угадаю кто на ней изображен! \n"
                                     "/search -- пришли мне имя и фамилию человека, а я тебе пришлю его фотографию) \n"
                                     "/makeup -- сделает макияж."
                                     )
```

Функция quiz

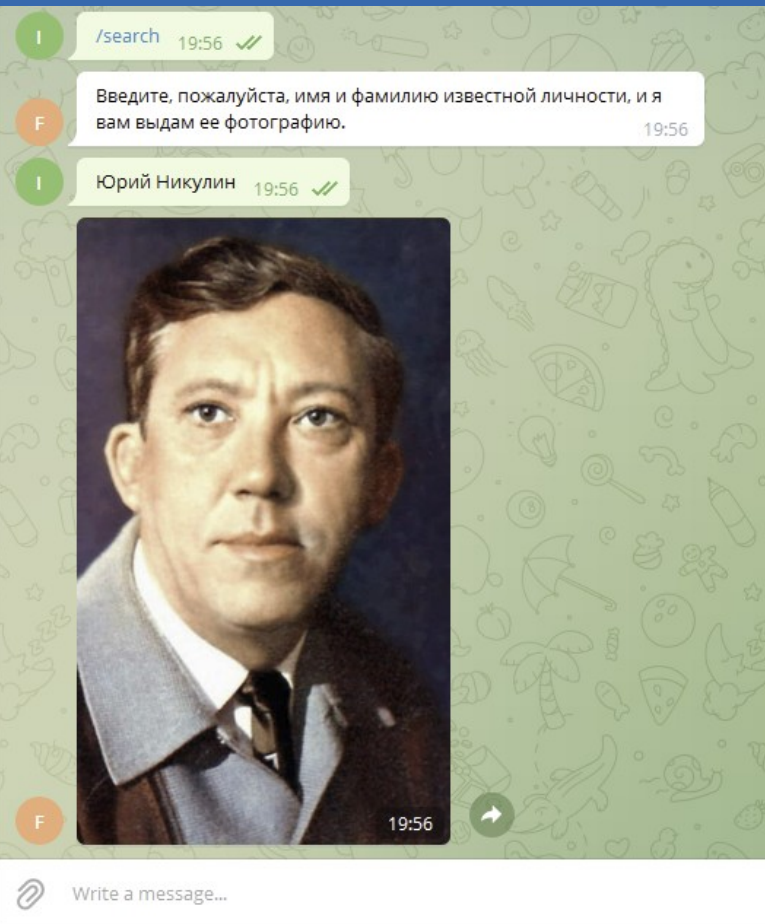


```
async def quiz(update, context):  
    """Отправляет сообщение когда получена команда /quiz"""  
    global parametr  
    parametr = "quiz"  
    await update.message.reply_text("Угадай личность! вопросов будет 3")  
    await question(update, context)
```

```
def making_quiz_photos():  
    global a  
    a = random.randint(0, len(list_of_famous_people))  
    url_photo = list_of_famous_people[a].split(", ")[1]  
    return url_photo  
  
Dyachkova  
async def question(update, context):  
    global count  
    url_photo = making_quiz_photos()  
    if parametr != "quiz3":  
        await update.message.reply_photo(photo=url_photo)  
    else:  
        await update.message.reply_text("Викторина окончена! Спасибо за участие! Вы ответили верно " + str(count) + " вопросов из 3")
```

Код для функции quiz

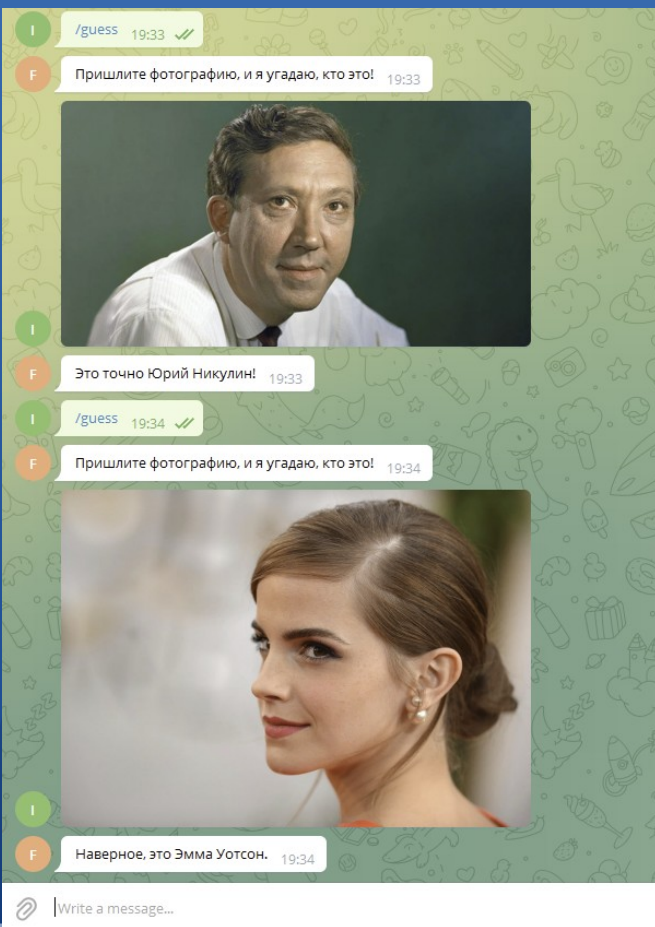
Функция search



```
async def search(update, context):  
    global parametr  
    parametr = "search"  
    await update.message.reply_text("Введите, пожалуйста, имя и фамилию известной личности, и я вам выдам ее фотографию.")  
  
    # Dyachkova  
    async def searching_photo(update, context):  
        user_text = update.message.text  
        if user_text in list_of_famous_people:  
            for i in range(len(list_of_famous_people)):  
                if user_text == list_of_famous_people[i].split(", ")[0]:  
                    url_photo = list_of_famous_people[i].split(", ")[1]  
                    await update.message.reply_photo(photo=url_photo)  
        else:  
            await update.message.reply_text("Извините, такого человека я не знаю...")
```

Код для функции search

Функция guess



```
async def guess(update, context):  
    global parametr  
    parametr = "guess"  
    await update.message.reply_text("Пришлите фотографию, и я угадаю, кто это!")
```

Код для функции guess

```
async def loading_picture(update, context):  
    obj = await context.bot.getFile(update.message.photo[-1].file_id)  
    file = get(obj.file_path)  
    print("Get user image " + str(file))  
    with open('mytmp.jpg', 'wb') as f:  
        f.write(file.content)  
    img_file = open("mytmp.jpg", "rb")  
    fc_image = face_recognition.load_image_file(img_file)  
    return fc_image  
  
# Dyachkova  
async def getting_photo(update, context):  
    global parametr  
    if parametr == "guess":  
        fc_image = await loading_picture(update, context)  
        face_encoding = face_recognition.face_encodings(fc_image)[-1]  
        face_distances = face_recognition.face_distance(known_encodings, face_encoding)  
        face_distance = face_distances.min()  
        count = 0  
        for i in range(len(face_distances)):  
            count += 1  
            if face_distances[i] == face_distance:  
                break  
        name = list_of_famous_people[count - 1].split(", ")[0]  
        if face_distance < 0.5:  
            await update.message.reply_text("Это точно " + name + "!")  
        elif face_distance > 0.5 and face_distance < 0.59:  
            await update.message.reply_text("Наверное, это " + name + ".")  
        else:  
            await update.message.reply_text("Я не знаю кто это(("  
            print("face distance of " + name + " is " + str(face_distance))  
    elif parametr == "makeup":  
        await making_up_photo(update, context)  
    else:
```


Функция makeup



```
async def making_up_photo(update, context):
    image = await loading_picture(update, context)
    face_landmarks_list = face_recognition.face_landmarks(image)
    print("LANDMARK_LIST -- ", face_landmarks_list)

    pil_image = Image.fromarray(image)
    for face_landmarks in face_landmarks_list:
        d = ImageDraw.Draw(pil_image, 'RGBA')

        # Make the eyebrows into a nightmare
        d.polygon(face_landmarks['left_eyebrow'], fill=eyebrows_color)
        d.polygon(face_landmarks['right_eyebrow'], fill=eyebrows_color)
        d.line(face_landmarks['left_eyebrow'], fill=eyebrows_color, width=2)
        d.line(face_landmarks['right_eyebrow'], fill=eyebrows_color, width=2)

        # Gloss the lips
        d.polygon(face_landmarks['top_lip'], fill=lips_color)
        d.polygon(face_landmarks['bottom_lip'], fill=lips_color)
        d.line(face_landmarks['top_lip'], fill=lips_color, width=2)
        d.line(face_landmarks['bottom_lip'], fill=lips_color, width=2)

        # Sparkle the eyes
        d.polygon(face_landmarks['left_eye'], fill=eyes_color)
        d.polygon(face_landmarks['right_eye'], fill=eyes_color)

        # Apply some eyeliner
        d.line(face_landmarks['left_eye'] + [face_landmarks['left_eye'][0]], fill=eyeliner_color, width=eyeliner_width)
        d.line(face_landmarks['right_eye'] + [face_landmarks['right_eye'][0]], fill=eyeliner_color, width=eyeliner_width)

    pil_image.save("mytmp1.jpg")
    await update.message.reply_photo(photo="mytmp1.jpg")
```

Код для функции makeup

Конец

Спасибо за внимание!