Чат-бот – Faceру

Автор – Дьячкова Дарья.

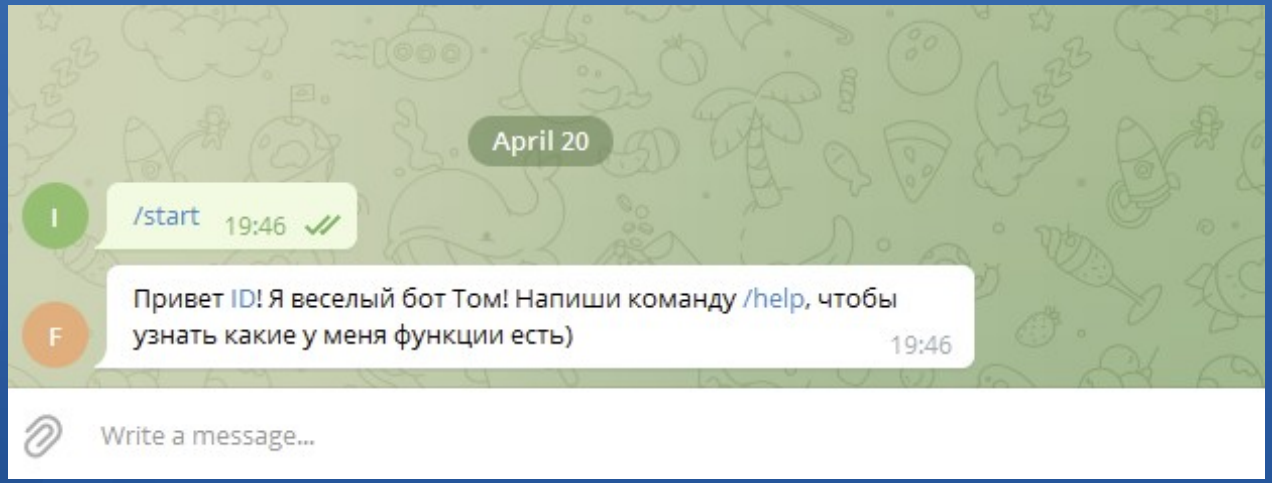
О проекте

Мой проект представляет собой чат-бот под названием Facepy. 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.

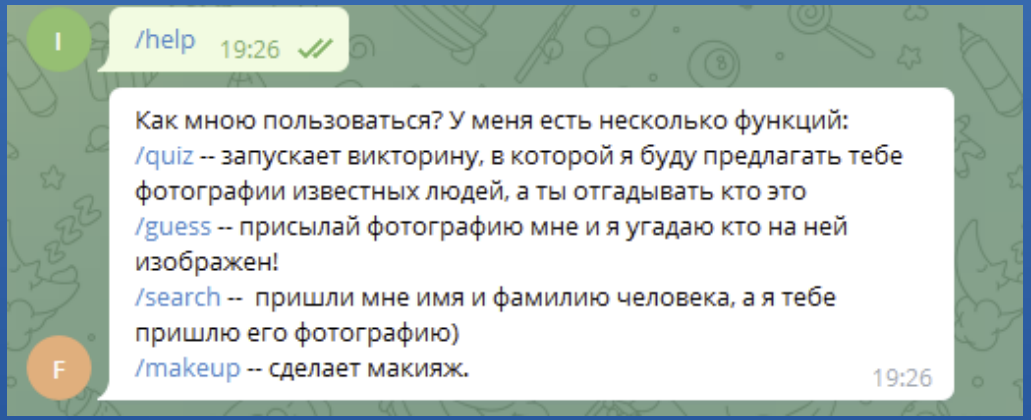
Функция start



Код для функции start

```python
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

```python
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."
                                    )
```

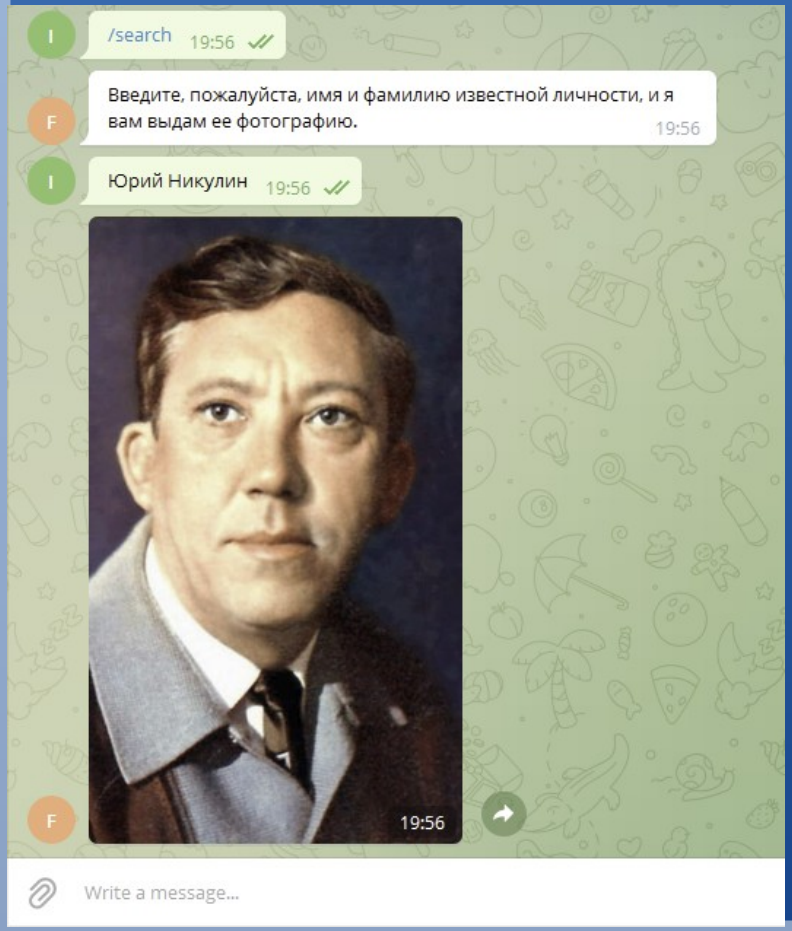
Функция quiz


```python
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)
```

```python
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("Вкиторина окончена!) Спасибо за участие! Вы ответили верно " + str(count) + " вопросов из 3")
```

Код для функции quiz

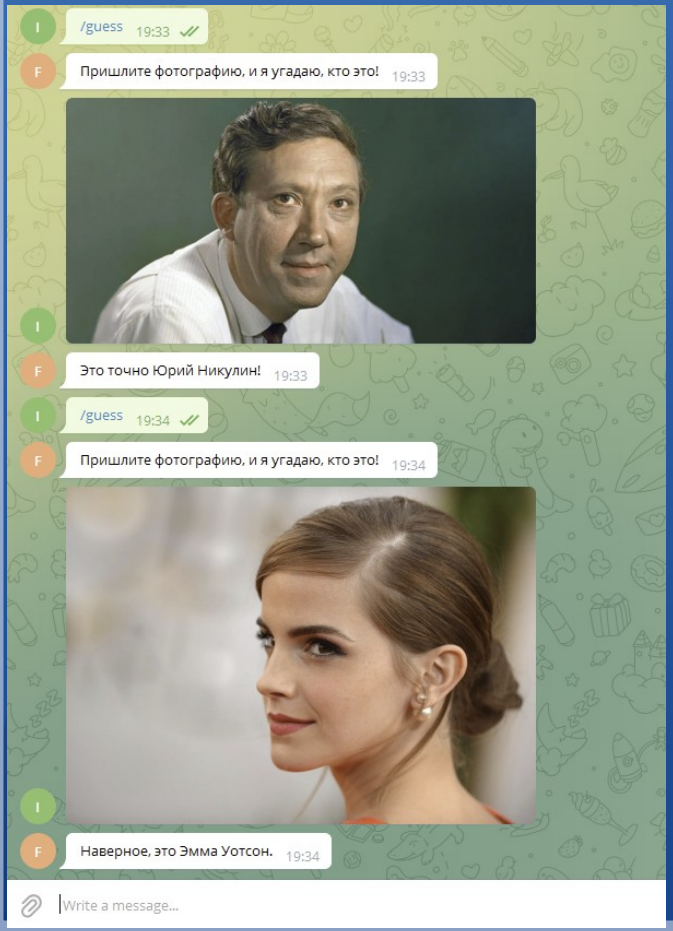
Функция search




```python
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




```python
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

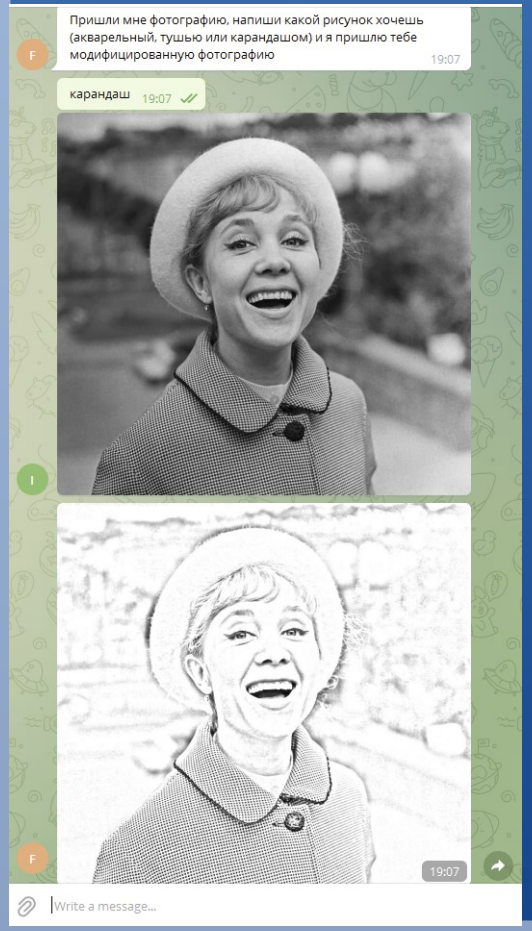
```python
async def loading_picture(update, context):
    obj = await context.bot.getFile(update.message.photo[-1].file_id)
    file = get(obj.file_path)
    print("Get user image " + str(file))
    with open('mytmp.jpg', 'wb') as f:
        f.write(file.content)
    img_file = open("mytmp.jpg", "rb")
    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


```python
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

Функция sketching




```python
async def sketching(update, context):
    global parametr
    parametr = "sketching"
    await update.message.reply_text("Пришли мне фотографию, напиши какой рисунок хочешь (акварельный, тушью или карандашом) и я пришлю тебе модифицированную фотогр

async def making_sketching(update, context):
    global picture
    await loading_picture(update, context)
    img = cv2.imread("mytmp.jpg")
    if picture == "акварель":
        # effect of watercolor image
        watercolour_image = cv2.stylization(img, sigma_s=10, sigma_r=0.45)
        cv2.imwrite('mytmp1.jpg', watercolour_image)
        await update.message.reply_photo(photo='mytmp1.jpg')
    elif picture == "карандаш":
        # effect of pencil sketch
        gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        img_invert = cv2.bitwise_not(gray_img)
        img_smoothing = cv2.GaussianBlur(img_invert, (25, 25), sigmaX=100, sigmaY=100)
        final_img = cv2.divide(gray_img, 255 - img_smoothing, scale=255)
        cv2.imwrite('mytmp1.jpg', final_img)
        await update.message.reply_photo(photo='mytmp1.jpg')
    elif picture == "тушь":
        # effect of ink
        gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        img_invert = cv2.bitwise_not(gray_img)
        img_smoothing = cv2.GaussianBlur(img_invert, (25, 25), sigmaX=100, sigmaY=100)
        final_img = cv2.divide(gray_img, 255 - img_smoothing, scale=255)
        img_smoothing = cv2.GaussianBlur(final_img, (25, 25), sigmaX=1, sigmaY=0.1)
        cv2.imwrite('mytmp1.jpg', (20 * 255) / (256 - img_smoothing))
        await update.message.reply_photo(photo='mytmp1.jpg')
```

Код для функции makeup

Спасибо за внимание!