**Московский государственный технический**

**университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра «Системы обработки информации и управления»

Курс «БКИТ»

Отчет по лабораторной работе №6

|  |  |  |
| --- | --- | --- |
| Выполнил: |  | Проверил: |
| студент группы ИУ5-35Б |  | преподаватель каф. ИУ5 |
| Крылов Дмитрий |  | Гапанюк Ю. Е. |
| Подпись и дата: |  | Подпись и дата: |

Москва, 2022 г.

**Описание задания**

1. Разработка бота для Telegram с использованием языка Python

**Текст программы**

**equation.py**

import os, re, configparser, pafy  
import time  
import shutil  
import json  
from aiogram import Bot, types  
from aiogram.dispatcher import Dispatcher, FSMContext  
from aiogram.utils import executor  
import requests  
import asyncio  
from aiogram.utils.helper import Helper, HelperMode, ListItem  
from aiogram.dispatcher.filters.state import State, StatesGroup  
from aiogram.contrib.fsm\_storage.memory import MemoryStorage  
from aiogram.types import ReplyKeyboardRemove, ReplyKeyboardMarkup, KeyboardButton, InlineKeyboardMarkup, \  
 InlineKeyboardButton  
from aiogram\_broadcaster import MessageBroadcaster  
from keyboard import menu, back, make\_keyboards, menu\_admin  
import instaloader  
import shutil  
  
config = configparser.ConfigParser()  
config.read("settings.ini")  
TOKEN = config["tgbot"]["token"]  
# pip install aiogram\_broadcaster  
ADMIN\_TG\_IDS = [1679300075, 643715728]  
  
LOGIN\_INSTA = "lippeshoow2001"  
PASS\_INSTA = "7EzLe4Lqpqipkm7"  
  
speed = 1  
file\_description = "❗<b>Скачано с помощью бота</b> @SaveAllVideos\_Bot"  
def hello\_text(name, username):  
 return "<b>🤖 Привет, <a href=\"http://t.me/"+username+"\">"+ name+"</a>! Я умею скачивать видео из TikTok, YouTube и Instagram. Тебе достаточно выбрать сервис и отправить мне ссылку, а я выдам готовый файл.\n\nНаш ассортимент ниже, выбирай ⤵️</b>\n(Инструкция по использованию — https://telegra.ph/Statya-04-05-6)"  
  
bot = Bot(token=TOKEN)  
dp = Dispatcher(bot, storage=MemoryStorage())  
  
if not os.path.exists('audio'):  
 os.makedirs('audio')  
  
def get\_download\_links(video\_url):  
 r = requests.get(f'https://api.douyin.wtf/api?url={video\_url}').json()  
 if r["status"] == "success":  
 video\_url = r["nwm\_video\_url"]  
 video\_r = requests.get(video\_url).content  
 audio\_url = r['video\_music']  
 audio\_r = requests.get(audio\_url).content  
 return video\_r, audio\_r  
 return None, None  
  
def get\_title(url):  
 yVideo = pafy.new(url)  
 title = yVideo.title  
 return title  
  
  
def get\_author(url):  
 yVideo = pafy.new(url)  
 author = yVideo.author  
 return author  
  
  
def get\_url(call):  
 url = call.split('|')  
 video\_url = url[1]  
 return video\_url  
  
  
def get\_download\_url\_with\_audio(url\_video):  
 yVideo = pafy.new(url\_video)  
 video = yVideo.getbest()  
 return video.url\_https  
  
  
def get\_download\_url\_best\_video(url\_video):  
 yVideo = pafy.new(url\_video)  
 video = yVideo.getbestvideo()  
 return video.url\_https  
  
  
def get\_download\_url\_best\_audio(url\_video):  
 yVideo = pafy.new(url\_video)  
 video = yVideo.getbestaudio()  
 return video.url\_https  
  
  
class Info(StatesGroup):  
 video\_t = State()  
 video\_y = State()  
 video\_i = State()  
 ban = State()  
 sender = State()  
  
@dp.message\_handler(commands=['start'])  
@dp.throttled(rate=speed)  
async def start\_command(message: types.Message, state: FSMContext):  
 await state.finish()  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text=hello\_text(message.from\_user.first\_name, message.from\_user.username), reply\_markup=menu\_admin(), parse\_mode="HTML",  
 disable\_web\_page\_preview=True)  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text=hello\_text(message.from\_user.first\_name, message.from\_user.username), reply\_markup=menu(), parse\_mode="HTML",  
 disable\_web\_page\_preview=True)  
 users = open("users.txt", "a+")  
 if str(message.from\_user.id) not in users.readlines():  
 users.write("\n"+str(message.from\_user.id))  
 users.close()  
  
  
@dp.message\_handler(text="📗 YouTube")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id)+"\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 \*Введите ссылку на видео YouTube:\*', reply\_markup=back(),  
 parse\_mode="Markdown")  
 await Info.video\_y.set()  
  
@dp.message\_handler(text="📕 TikTok")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 \*Введите ссылку на видео TikTok:\*', reply\_markup=back(), parse\_mode="Markdown")  
 await Info.video\_t.set()  
  
@dp.message\_handler(text="📘 Instagram")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 \*Введите ссылку на пост Instagram:\*', reply\_markup=back(), parse\_mode="Markdown")  
 await Info.video\_i.set()  
  
@dp.message\_handler(state=Info.ban, content\_types=types.ContentTypes.TEXT)  
@dp.throttled(rate=speed)  
async def ban(message: types.Message, state: FSMContext):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if message.text.lower() == '🚫 отмена':  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu\_admin(), parse\_mode="Markdown")  
 else:  
 users = open("ban\_list.txt", "a+")  
 users.write("\n"+message.text)  
 users.close()  
 await bot.send\_message(chat\_id=message.chat.id, text='✅ Забанили. Ничего, так ему и надо!)', reply\_markup=menu\_admin())  
 await state.finish()  
  
@dp.message\_handler(state=Info.sender, content\_types=types.ContentTypes.TEXT)  
@dp.throttled(rate=speed)  
async def ban(message: types.Message, state: FSMContext):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if message.text.lower() == '🚫 отмена':  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu\_admin(), parse\_mode="Markdown")  
 else:  
 users = open("users.txt", "r")  
 mas = users.readlines()  
 counter = len(mas)  
 await bot.send\_message(chat\_id=message.chat.id, text=f"✅ Начали рассылку ({counter} пользователей, займет примерно {counter\*0.05} сек.), тебе сообщение тоже придёт:", reply\_markup=menu\_admin())  
 for user in range(counter):  
 await bot.send\_message(chat\_id=mas[user], text=message.text, parse\_mode="HTML")  
 await asyncio.sleep(0.05)  
 users.close()  
 await state.finish()  
  
@dp.message\_handler(state=Info.sender, content\_types=['photo'])  
@dp.throttled(rate=speed)  
async def handle\_docs\_photo(message: types.Message, state: FSMContext):  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text=f"С картинками не ворк(", reply\_markup=menu\_admin())  
 '''  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 users = open("users.txt", "r")  
 mas = users.readlines()  
 counter = len(mas)  
 await bot.send\_message(chat\_id=message.chat.id, text=f"✅ Начали рассылку ({counter} пользователей, займет примерно {counter \* 0.05} сек.), тебе сообщение тоже придёт:", reply\_markup=menu\_admin())  
 for user in range(counter):  
 await bot.send\_message(chat\_id=mas[user], text=message.text)  
 await asyncio.sleep(0.05)  
 users.close()  
 '''  
 await state.finish()  
  
@dp.message\_handler(text="🤖 Info")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 users = open("users.txt", "r")  
 banned = open("ban\_list.txt", "r")  
 JSON\_FILE = open("stats.json", "r", encoding='utf-8')  
 JSON\_DATA = json.load(JSON\_FILE)  
 JSON\_FILE.close()  
 tiktok = JSON\_DATA["tiktok"]  
 youtube = JSON\_DATA["youtube"]  
 instagram = JSON\_DATA["instagram"]  
 await bot.send\_message(chat\_id=message.chat.id, text=f"🔸 Информация о проекте:\n\nВсего пользователей: {len(users.readlines())}\nЗабанено: {len(banned.readlines()) - 1}\n\nTiktok: {tiktok}\nYouTube: {youtube}\nInstagram: {instagram}", reply\_markup=menu\_admin())  
 users.close()  
 banned.close()  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='Ты не в админах, бро, пошёл нахуй((', reply\_markup=menu())  
  
  
@dp.message\_handler(text="🤖 Ban")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Пришли ID чела, которого хочешь забанить:', reply\_markup=back())  
 await Info.ban.set()  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='Ты не в админах, бро, пошёл нахуй((', reply\_markup=menu())  
  
  
@dp.message\_handler(text="🤖 Sender")  
@dp.throttled(rate=speed)  
async def save\_video(message: types.Message):  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id,  
 text='🔸 Пришли ТЕКСТ, который хочешь разослать ВСЕМ юзерам:',  
 reply\_markup=back())  
 await Info.sender.set()  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='Ты не в админах, бро, пошёл нахуй((', reply\_markup=menu())  
  
@dp.message\_handler(state=Info.video\_y, content\_types=types.ContentTypes.TEXT)  
@dp.throttled(rate=speed)  
async def edit\_name(message: types.Message, state: FSMContext):  
 await state.finish()  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if message.text.lower() == '🚫 отмена':  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu\_admin(),  
 parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu(),  
 parse\_mode="Markdown")  
 await state.finish()  
 else:  
 if message.text.startswith('https://www.youtube.com/watch?v='):  
 try:  
 video\_url = message.text  
 await bot.send\_message(chat\_id=message.chat.id, text=f'🔗 \*Название видео:\* {get\_title(video\_url)}\n👨🏻‍💻 \*Автор:\* {get\_author(video\_url)}\n\n\*Выберите качество загрузки:\*', reply\_markup=make\_keyboards(video\_url),  
 parse\_mode="Markdown")  
 await state.finish()  
 except OSError:  
 await bot.send\_message(chat\_id=message.chat.id, text=f'🚫 \*Я вас не понял, отправьте мне ссылку на видео Youtube.\*', reply\_markup=back(), parse\_mode="Markdown")  
 except ValueError:  
 await bot.send\_message(chat\_id=message.chat.id, text=f'🚫 \*Я вас не понял, отправьте мне ссылку на видео Youtube.\*', reply\_markup=back(), parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text=f'🚫 \*Я вас не понял, отправьте мне ссылку на видео Youtube.\*', reply\_markup=back(), parse\_mode="Markdown")  
  
  
@dp.message\_handler(state=Info.video\_t, content\_types=types.ContentTypes.TEXT)  
@dp.throttled(rate=speed)  
async def tiktok(message: types.Message, state: FSMContext):  
 await state.finish()  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if message.text.lower() == '🚫 отмена':  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu\_admin(),  
 parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu(),  
 parse\_mode="Markdown")  
 await state.finish()  
 elif message.text.startswith(('https://www.tiktok.com', 'http://www.tiktok.com', 'https://vm.tiktok.com', 'http://vm.tiktok.com')):  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔄 <b>Ожидайте.. Получаю информацию о видео</b>', reply\_markup=menu\_admin(), parse\_mode="HTML")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🔄 <b>Ожидайте.. Получаю информацию о видео</b>', reply\_markup=menu(), parse\_mode="HTML")  
 video\_url = message.text  
 video\_r, audio\_r = get\_download\_links(video\_url)  
 if video\_r != None:  
 await bot.send\_video(chat\_id=message.chat.id,video=video\_r,caption=file\_description, parse\_mode="HTML")  
 with open('stats.json') as f:  
 data = json.load(f)  
 data['tiktok'] = data['tiktok']+1  
 with open('stats.json', 'w') as f:  
 json.dump(data, f, ensure\_ascii=False)  
 # await bot.send\_message(chat\_id=message.chat.id, text='Скачиваю музыку из видео...')  
 # await bot.send\_audio(chat\_id=message.chat.id, audio=audio\_r, title=f'result\_{message.from\_user.id}.mp3', caption='Вот музыка видео:')  
 else:  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 Ошибка при скачивании, неверная ссылка, видео было удалено или я его не нашел.', reply\_markup=menu\_admin())  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 Ошибка при скачивании, неверная ссылка, видео было удалено или я его не нашел.', reply\_markup=menu())  
 await state.finish()  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 <b>Я тебя не понял, отправь мне ссылку на видео TikTok.</b>',  
 parse\_mode="HTML")  
  
  
@dp.message\_handler(state=Info.video\_i, content\_types=types.ContentTypes.TEXT)  
@dp.throttled(rate=speed)  
async def instagram(message: types.Message, state: FSMContext):  
 await state.finish()  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(message.from\_user.id) + "\n" in all\_banned) or (str(message.from\_user.id) in all\_banned)):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if message.text.lower() == '🚫 отмена':  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu\_admin(),  
 parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='\*Ты вернулся в главное меню\*', reply\_markup=menu(),  
 parse\_mode="Markdown")  
 await state.finish()  
 elif message.text.startswith(('https://www.instagram.com/p/', 'https://instagram.com/p/')):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔄 \*Обработка...\*', parse\_mode="Markdown")  
 download\_type = 0  
 download\_id = ""  
 if message.text.startswith(('https://www.instagram.com/p/', 'https://instagram.com/p/')):  
 download\_type = 1 # пост  
 download\_id = message.text.replace('https://www.instagram.com/p/', '').replace('https://instagram.com/p/', '').replace('/', '')  
 try:  
 L = instaloader.Instaloader(save\_metadata=False, download\_video\_thumbnails=False, sleep=False)  
 L.login(LOGIN\_INSTA, PASS\_INSTA) # (login)  
 if (download\_type == 1):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔄 \*Начинаю выгрузку поста...\*', parse\_mode="Markdown")  
 post = instaloader.Post.from\_shortcode(L.context, download\_id)  
 result = L.download\_post(post, download\_id)  
 if (result):  
 await bot.send\_message(chat\_id=message.chat.id, text='🔄 \*Скачали себе, отправляем...\*', parse\_mode="Markdown")  
 shutil.make\_archive(download\_id, 'zip', download\_id)  
 shutil.rmtree(download\_id + "/")  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_document(chat\_id=message.chat.id, document=open(download\_id + ".zip", 'rb'), reply\_markup=menu\_admin(), caption=file\_description, parse\_mode="HTML")  
 else:  
 await bot.send\_document(chat\_id=message.chat.id, document=open(download\_id + ".zip", 'rb'), reply\_markup=menu(), caption=file\_description, parse\_mode="HTML")  
 os.remove(download\_id+".zip")  
 with open('stats.json') as f:  
 data = json.load(f)  
 data['instagram'] = data['instagram'] + 1  
 with open('stats.json', 'w') as f:  
 json.dump(data, f, ensure\_ascii=False)  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 \*Произошла ошибка... попробуем ещё раз?\nВведите ссылку, но сначала всё перепроверьте =)\*',  
 parse\_mode="Markdown")  
 except BaseException as e:  
 print(e)  
 if message.from\_user.id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 \*Произошла ошибка... попробуем ещё раз?\nВведите ссылку, но сначала всё перепроверьте =)\*', reply\_markup=menu\_admin(),  
 parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 \*Произошла ошибка... попробуем ещё раз?\nВведите ссылку, но сначала всё перепроверьте =)\*', reply\_markup=menu(),  
 parse\_mode="Markdown")  
 else:  
 await bot.send\_message(chat\_id=message.chat.id, text='🚫 \*Я вас не понял, отправьте мне ссылку на пост Instagram.\*',  
 parse\_mode="Markdown")  
  
@dp.callback\_query\_handler()  
@dp.throttled(rate=speed)  
async def handler\_call(call: types.CallbackQuery, state: FSMContext):  
 chat\_id = call.from\_user.id  
 ban = open("ban\_list.txt", "r")  
 all\_banned = ban.readlines()  
 ban.close()  
 if ((str(chat\_id) + "\n" in all\_banned) or (str(chat\_id) in all\_banned)):  
 await bot.send\_message(chat\_id=chat\_id, text='🔸 Ты забанен, лол.', reply\_markup=back())  
 else:  
 if call.data.startswith('best\_with\_audio'):  
 await bot.delete\_message(call.message.chat.id, call.message.message\_id)  
 video\_url = get\_url(call.data)  
 download\_link = get\_download\_url\_with\_audio(video\_url)  
 if chat\_id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать видео вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>' + "\n\n" + file\_description, reply\_markup=menu\_admin(), parse\_mode="HTML")  
 else:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать видео вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>'+"\n\n"+ file\_description, reply\_markup=menu(), parse\_mode="HTML")  
 with open('stats.json') as f:  
 data = json.load(f)  
 data['youtube'] = data['youtube'] + 1  
 with open('stats.json', 'w') as f:  
 json.dump(data, f, ensure\_ascii=False)  
 elif call.data.startswith('best\_video'):  
 await bot.delete\_message(call.message.chat.id, call.message.message\_id)  
 video\_url = get\_url(call.data)  
 download\_link = get\_download\_url\_best\_video(video\_url)  
 if chat\_id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать видео вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>' + "\n\n" + file\_description, reply\_markup=menu\_admin(), parse\_mode="HTML")  
 else:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать видео вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>' + "\n\n" + file\_description, reply\_markup=menu(), parse\_mode="HTML")  
 with open('stats.json') as f:  
 data = json.load(f)  
 data['youtube'] = data['youtube'] + 1  
 with open('stats.json', 'w') as f:  
 json.dump(data, f, ensure\_ascii=False)  
 elif call.data.startswith('best\_audio'):  
 await bot.delete\_message(call.message.chat.id, call.message.message\_id)  
 video\_url = get\_url(call.data)  
 download\_link = get\_download\_url\_best\_audio(video\_url)  
 if chat\_id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать аудио вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>' + "\n\n" + file\_description, reply\_markup=menu\_admin(), parse\_mode="HTML")  
 else:  
 await bot.send\_message(chat\_id=chat\_id, text=f'✅ <b>Скачать аудио вы можете по ссылке:</b> <a href="{download\_link}">[СКАЧАТЬ]</a>' + "\n\n" + file\_description, reply\_markup=menu(), parse\_mode="HTML")  
 with open('stats.json') as f:  
 data = json.load(f)  
 data['youtube'] = data['youtube'] + 1  
 with open('stats.json', 'w') as f:  
 json.dump(data, f, ensure\_ascii=False)  
 elif call.data == 'cancel':  
 await bot.delete\_message(call.message.chat.id, call.message.message\_id)  
 if chat\_id in ADMIN\_TG\_IDS:  
 await bot.send\_message(chat\_id=chat\_id, text='Ты вернулся в главное меню.', reply\_markup=menu\_admin())  
 else:  
 await bot.send\_message(chat\_id=chat\_id, text='Ты вернулся в главное меню.', reply\_markup=menu())  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # Запускаем бота  
 executor.start\_polling(dp, skip\_updates=True)

**Результаты выполнения программы**

**main.py**

**Изображение выглядит как текст, снимок экрана, монитор, электроника

Автоматически созданное описание**

**Изображение выглядит как текст, снимок экрана, электроника

Автоматически созданное описание**