

**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:

студент группы ИУ5-35Б
Крылов Дмитрий

Подпись и дата:

Проверил:

преподаватель каф. ИУ5
Гапанюк Ю. Е.

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()

@dpm.message_handler(commands=['start'])
@dpm.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.ContentType.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'👁 *Я
*Название видео:* {get_title(video_url)}\n📺 *Автор:*
{get_author(video_url)}\n\n*Выберите качество загрузки:*\n',
                    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")

@dps.message_handler(state=Info.video_t, content_types=types.ContentTypes.TEXT)
@dps.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

