Московский авиационный институт

(национальный исследовательский университет)

Факультет № 8 «Информационные технологии и прикладная математика»Кафедра 806 «Вычислительная математика и программирование»

РЕФЕРАТ

по дисциплине «Фундаментальная информатика»

1 семестр

на тему “Телеграмм-бот, который отправляет актуальную погоду в городе”

Студент:	Шамбилов Р.Т.
Группа:	М8О-109Б-22
Преподаватель:	Сысоев М.А.
Подпись:	
Оценка:	

Москва, 2022

СОДЕРЖАНИЕ

1.ПЛАН ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучение библиотеки, которая позволяет создать собственного бота в телеграмм
2. Изучение команд библиотеки
3. Изучение библиотеки requests, которая будет кидать запросы на сайт
4. Изучение библиотеки datetime, которая будет переводить время в стандартный вид (часы, минуты, секунды)
5. Изучение некоторых команд Python (try, except)
6. Тестирование бота в самом телеграмме

2. НАПИСАНИЕ ПРОГРАММЫ

1. telebot (pyTelegramBotAPI) хорошая и лёгкая библиотека для создания бота на python для телеграмма.

Благодаря данной библиотеки мы можем взаимодействовать с заранее созданным ботом в тг (`bot = telebot.TeleBot(tg_bot_token)`

`@bot.message_handler(commands=['start'])` - команда для создания кнопки старт, с которой бот начнет работать

`@bot.message_handler(commands=['help'])` - то же самое, что и /start, только при прописке /help бот повторит, что нужно сделать, чтобы взаимодействовать с ним

`@bot.message_handler()` - бот будет обращаться к этой команде при любых сообщениях

2. Библиотека requests является стандартным инструментом для составления HTTP-запросов в Python. Простой и аккуратный API значительно облегчает трудоемкий процесс создания запросов. Таким образом, можно сосредоточиться на взаимодействии со службами и использовании данных в приложении.

Благодаря данной библиотеки мы будем отсылать запросы на сайт

```
r = requests.get(f"https://api.openweathermap.org/data/2.5/weather?q={message.text}&appid={open_weather_token}&units=metric")
```

Таким образом мы отсылаем наш запрос на сайт, который выдаст нам сводку по материалу. Так же стоит упомянуть команду `data = r.json()`

Которая позволит нам собраться всю информацию.

Теперь по коду:

Сам код будет писать в одну функцию, которая отслеживает введеный город при помощи команды try, если она не распознает никакой город будет использоваться - except.

```
try:
    r = requests.get(f"https://api.openweathermap.org/data/2.5/weather?q={message.text}&appid={open_weather_token}&units=metric")
    data = r.json()
```

Посылаем запрос на сайт - получаем ответ - запоминаем, полученную информацию

```
city = data["name"]
current_temp = data["main"]["temp"]
max_temp = data["main"]["temp_max"]
min_temp = data["main"]["temp_min"]

weather_description = data["weather"][0]["main"]
if weather_description in code_to_smile:
    sm = code_to_smile[weather_description]
else:
    sm = "Посмотри в окно, не пойму, что там за погода!"

humidity = data["main"]["humidity"]
wind = data["wind"]["speed"]
sunrise_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunrise"])
sunset_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunset"])
length_of_the_day = datetime.datetime.fromtimestamp(data["sys"]["sunset"]) - \
                    datetime.datetime.fromtimestamp(data["sys"]["sunrise"])
```

Здесь мы отслеживаем специальные ключи, которые и помогут нам выдать разную информацию о погоде. Стоит выделить отдельное условия для отправки смайликов, которые так же отправляются при помощи ключей

```
code_to_smile = {
    "Clear": "Ясно ☀️",
    "Clouds": "Облачно ☁️",
    "Rain": "Дождь 🌧️",
    "Drizzle": "Дождь 🌧️",
    "Thunderstorm": "Гроза 🌩️",
    "Snow": "Снег \U0001F328",
    "Mist": "Туман \U0001F32B"
}
```

РАБОТА БОТА: Сам бот здесь работает только для улавливания сообщений и отправки данных, что реализуются при помощи следующих команд:

```
@bot.message_handler()
```
- бот улавливает все сообщения, отправленные пользователем

```
bot.send_message(message.chat.id, "🔴 <u>Проверьте название города</u> 🔴", parse_mode='html')
```
- в конкретно данной форме бот отправит сообщение об ошибке, так как город

был не найден.

- и последняя команда для того, чтобы бот работал, при запуске программы.

3.ЗАКЛЮЧЕНИЕ

Благодаря проделанной работе я получил большой опыт в работе с тг-ботами, тк сделал около 3-4 ботов, которые выполняют те или иные действия. Больше углубился в библиотеки питона, а также в сам язык.

В общем работа понравилась, на мой взгляд, это куда лучше, чем доклад о языках программирования, за что +реп Максу.