**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ (НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»**

**Журнал практики**

Студента *Кирюшкина Даниила Александровича*

##### Факультет Ступинский филиал МАИ

###### Кафедра *«Моделирование систем и информационные технологии»*

##### Учебная группа ТСО-105Б-22

Направление подготовки (специальность) *09.03.01*

*(шифр)*

*«Информатика и вычислительная техника»*

*(название направления, специальности)*

Вид практики *учебная*

*(учебной, производственной, преддипломной или другой вид практики)*

Руководитель практики от МАИ

*Мамонов Игорь Михайлович*  \_\_\_\_\_\_\_\_\_\_\_\_\_

*(фамилия, имя, отчество) (подпись)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/*Кирюшкин Д.А.*/ “12” июля 2023 г.

*(подпись студента)* *(дата)*

Ступино 2023

**1.Место и сроки проведения практики**

*Сроки проведения практики:*

*-дата начала практики 29.06.2023 г.*

*-дата окончания практики 12.07.2023 г.*

*Наименование предприятия Ступинский филиал МАИ*

*Название структурного подразделения (отдел, лаборатория) кафедра «Моделирование систем и информационные технологии»*

1. **Инструктаж по технике безопасности**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/Беспалов Б.Б./ “29” июня 2023 г.

*(подпись проводившего)* *(дата проведения)*

1. **Индивидуальное задание студенту**

Разработка чат бота, передающего погодные данные на языке python в сети Telegram

**4.План выполнения индивидуального задания**

1. Исследовать и проанализировать возможности чат-ботов.

2. Выбрать и загрузить язык разработки.

3. Выбрать среду и инструменты разработки.

4. Установить нужные компоненты и библиотеки.

4. Продумать функционал чат-бота.

5. Написать и протестировать чат-бота.

*Руководитель практики от МАИ*: */Мамонов И.М./*

*Руководитель от предприятия*:  */ /*

*/Кирюшкин Д.А./* “29” июня 2023 г.

*(подпись студента)* *(дата)*

**5.Отзыв руководителя практики от предприятия**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Руководитель от предприятия*: / /

*(подпись) (фамилия, имя, отчество)*

«\_\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_2023 г.

М.П. (печать)

**6.Отчет студента о практике**

Целью учебной практики являлась разработка бота, передающего пользователю краткую сводку основных необходимых в повседневной жизни данных о погоде в два нажатия.

Был использован язык программирования python, библиотеки os, math, telebot, config и requests, hson ; среда разработки VS code. Проект представляет собой чат с ботом, который, после активации запросит у вас необходимый город. После поступления этой информации, он, в свою очередь сообщит данные на текущий момент времени, такие как: температуру воздуха, влажность, давление и скорость ветра. Отдельно был оформлен внешний вид, который по задумке должен вызывать приятные эмоции и немного поднимать настроение. Приложение использует Python 3.7.

Код систематизирован и содержит комментарии, что позволяет в нем ориентироваться.

**Структура проекта**

Проект называется Бот Андюха(BotAndreuha)

Изображения:  
Из. 1 – Начальное диалоговое окно

Из. 2 – Первый пример работы

Из. 3 – Второй пример работы

**Основная** **часть**

Начнем с основного файла, bot.py:

Соответственно, в нём хранится основной код бота.

Перед началом работы импортируем все необходимые библиотеки:

import os

import math

import telebot

from telebot import types

import config

import requests, json

Затем соединим код с уникальным ключом-идентификатором бота через ссылку на файл, в который для удобства была помещена соответствующая информация.

bot = telebot.TeleBot(config.TELEGRAM\_API\_KEY)

Определим функцию и так же соединим с идентификатором с сайта openweathermap.com для дальнейшего получения точной и бесплатной информации в реальном времени. Теперь разберем каждую строку:

1. BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?" - создание переменной BASE\_URL и присвоение ей значения "https://api.openweathermap.org/data/2.5/weather?". Это базовый URL для запроса к API погоды.

2. URL = BASE\_URL + "q=" + city + "&appid=" + API\_KEY - создание переменной URL и присвоение ей значения, которое состоит из BASE\_URL, строки "q=", значения переменной city, строки "&appid=" и значения переменной API\_KEY. Это полный URL для запроса к API погоды, включающий город и ключ API.

3. response = requests.get(URL, params={'units': 'metric', 'lang': 'ru'}) - отправка GET-запроса по указанному URL с параметрами 'units': 'metric' (единицы измерения в метрической системе) и 'lang': 'ru' (язык ответа на русском). Ответ от сервера сохраняется в переменной response.

def weather(city, API\_KEY):

    BASE\_URL = "https://api.openweathermap.org/data/2.5/weather?"

    URL = BASE\_URL + "q=" + city + "&appid=" + API\_KEY

    response = requests.get(URL,

                            params={'units': 'metric', 'lang': 'ru'})

Проверив статус кода ответа от сервера, работаем дальше. После этого можно задать некоторые переменные, необходимые для дальнейшей работы, передачи и хранения информации.

if response.status\_code == 200:

        data = response.json()

        weather\_description = data["weather"][0]["main"]

        temperature = data["main"]["temp"]

        humidity = data["main"]["humidity"]

        pressure = data["main"]["pressure"]

        wind = data["wind"]["speed"]

        return temperature, humidity, pressure, wind

- if response.status\_code == 200: - проверка статус кода ответа от сервера. Если код равен 200, то выполняется следующий блок кода.

- data = response.json() - преобразование ответа от сервера в формат JSON и сохранение его в переменной data.

- weather\_description = data["weather"][0]["main"] - извлечение значения ключа "main" из первого элемента списка "weather" в переменную weather\_description. Это описание погоды.

- temperature = data["main"]["temp"] - извлечение значения ключа "temp" из словаря "main" в переменную temperature. Это температура.

- humidity = data["main"]["humidity"] - извлечение значения ключа "humidity" из словаря "main" в переменную humidity. Это влажность.

- pressure = data["main"]["pressure"] - извлечение значения ключа "pressure" из словаря "main" в переменную pressure. Это давление.

- wind = data["wind"]["speed"] - извлечение значения ключа "speed" из словаря "wind" в переменную wind. Это скорость ветра.

- return temperature, humidity, pressure, wind - возврат значений температуры, влажности, давления и скорости ветра из функции.

В случае, если запрос некорректен, дождёмся корректного запроса. Было принято решение не раздражать пользователя запросами повторить запрос.

else:

        print()

Затем займемся работой бота непосредственно в telegram. Для начала назначим активацию бота на классическую команду /start. Затем запустим функцию, отвечающую за приветствие, объяснение принципа работы и передачу информации. Далее подробнее.

@bot.message\_handler(commands=['start'])

def start\_message(message):

    msg = bot.reply\_to(message, f"Здравствуйте, <b><u>{message.from\_user.username}</u></b>! Я - бот показывающий текущую погоду \U0001F31E по городу.", parse\_mode='html')

    bot.send\_message(message.chat.id, "Введите название города, чтобы продолжить")

    bot.register\_next\_step\_handler(msg, print\_weather)

- @bot.message\_handler(commands=['start']) - декоратор, который указывает, что функция start\_message будет вызываться при получении команды /start от пользователя.

- def start\_message(message): - объявление функции start\_message, которая принимает параметр message.

- msg = bot.reply\_to(message, f"Здравствуйте, <b><u>{message.from\_user.username}</u></b>! Я - бот показывающий текущую погоду \U0001F31E по городу.", parse\_mode='html') - отправка ответного сообщения пользователю с приветствием и его именем. Результат сохраняется в переменной msg.

- bot.send\_message(message.chat.id, "Введите название города, чтобы продолжить") - отправка сообщения пользователю с просьбой ввести название города.

- bot.register\_next\_step\_handler(msg, print\_weather) - регистрация следующего шага обработки сообщения от пользователя с помощью функции print\_weather.

def print\_weather(message):

    city = message.text

    temperature = weather(city, config.WEATHER\_API\_KEY)

    humidity = weather(city, config.WEATHER\_API\_KEY)

    pressure = weather(city, config.WEATHER\_API\_KEY)

    wind = weather(city, config.WEATHER\_API\_KEY)

    msg = bot.reply\_to(message, f"Текущая температура: {temperature[0]}°C\nВлажность: {humidity[1]}%\nДавление: {pressure[2]} мм.рт.ст\nВетер: {wind[3]} м/с")

    bot.send\_message(message.chat.id, "Введите название города, чтобы продолжить")

    bot.register\_next\_step\_handler(msg, print\_weather)

bot.polling()

В заключение, разберемся с выводом информации и создадим замкнутый цикл.

- def print\_weather(message): - объявление функции print\_weather, которая принимает параметр message.

- city = message.text - сохранение текста сообщения пользователя в переменную city.

- temperature = weather(city, config.WEATHER\_API\_KEY) - вызов функции weather с параметрами city и config.WEATHER\_API\_KEY для получения текущей температуры.

- humidity = weather(city, config.WEATHER\_API\_KEY) - вызов функции weather с параметрами city и config.WEATHER\_API\_KEY для получения влажности.

- pressure = weather(city, config.WEATHER\_API\_KEY) - вызов функции weather с параметрами city и config.WEATHER\_API\_KEY для получения давления.

- wind = weather(city, config.WEATHER\_API\_KEY) - вызов функции weather с параметрами city и config.WEATHER\_API\_KEY для получения скорости ветра.

- msg = bot.reply\_to(message, f"Текущая температура: {temperature[0]}°C\nВлажность: {humidity[1]}%\nДавление: {pressure[2]} мм.рт.ст\nВетер: {wind[3]} м/с") - отправка ответного сообщения пользователю с текущей погодой (температурой, влажностью, давлением и скоростью ветра).

- bot.send\_message(message.chat.id, "Введите название города, чтобы продолжить") - отправка сообщения пользователю с просьбой ввести название следующего города.

- bot.register\_next\_step\_handler(msg, print\_weather) - регистрация следующего шага обработки сообщения от пользователя с помощью функции print\_weather.

- bot.polling() - запуск бота для получения и обработки новых сообщений.

Второй основной файл – **config.py**:

WEATHER\_API\_KEY = "c8d71d072e33a8bd48a31662ac0b22d3"

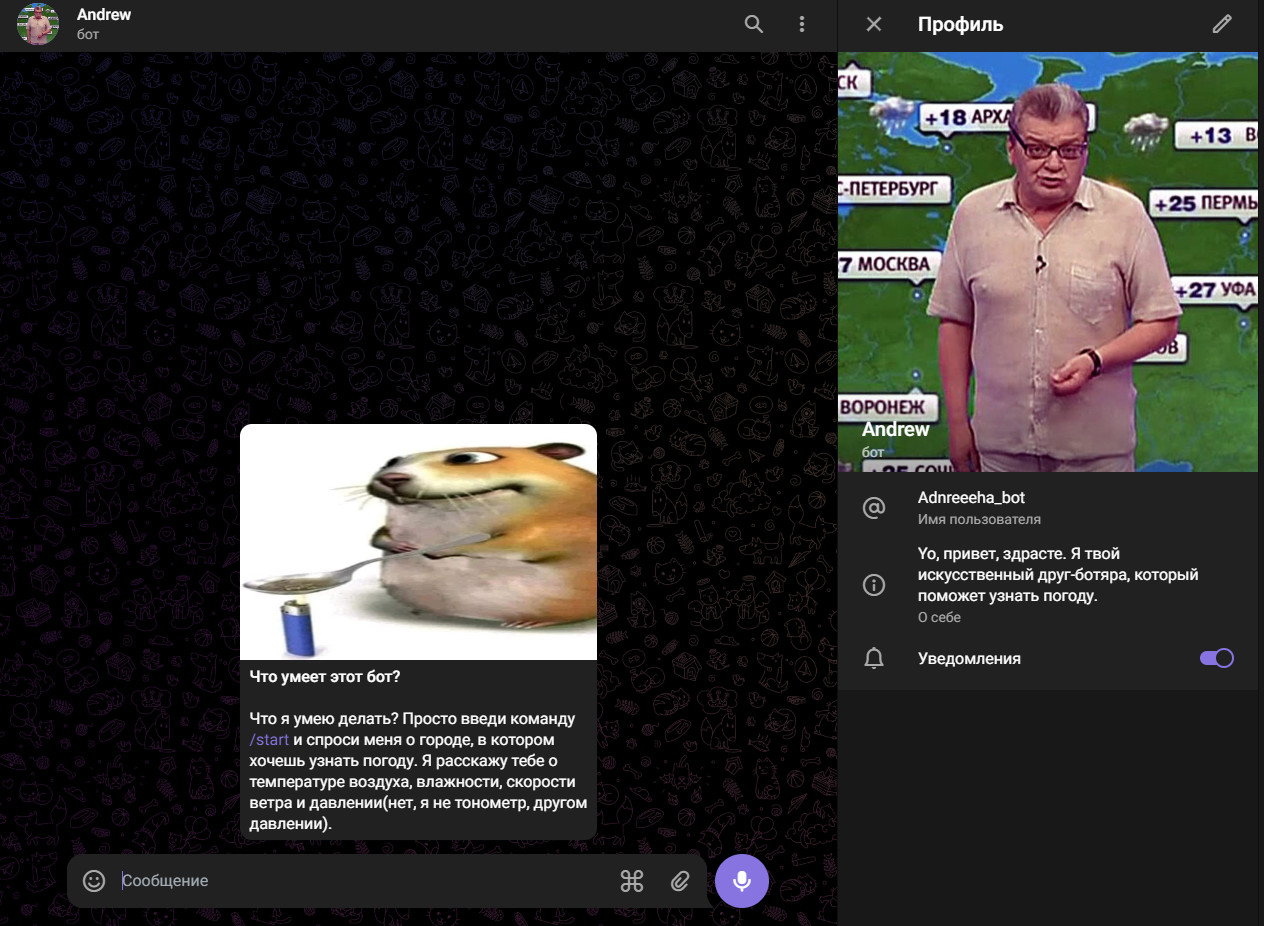
TELEGRAM\_API\_KEY = "6209891725:AAFvKM6Dgy4OUAqOw9Az3MD-jY7HU8lIG-M"

Просто содержит уникальные ключи. Создан для удобства.

Таким образом, можно перейти к **тестированию бота**. Все результаты теста далее в формате скриншотов.

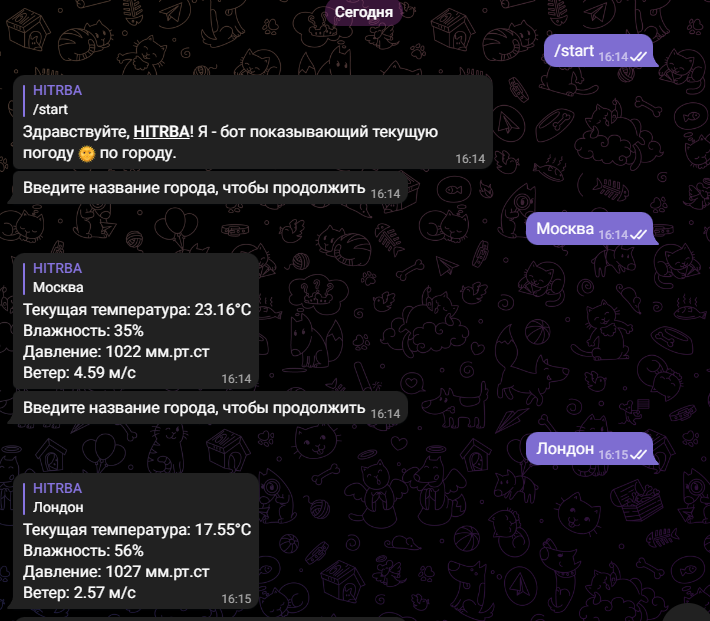
Описание написано довольно фривольно и должно вызвать смущение, смех и желание узнать, что же тут не так? Далее то, что увидит человек, впервые зайдя в диалоговое окно.

Из. 1 – Начальное иалоговое окно

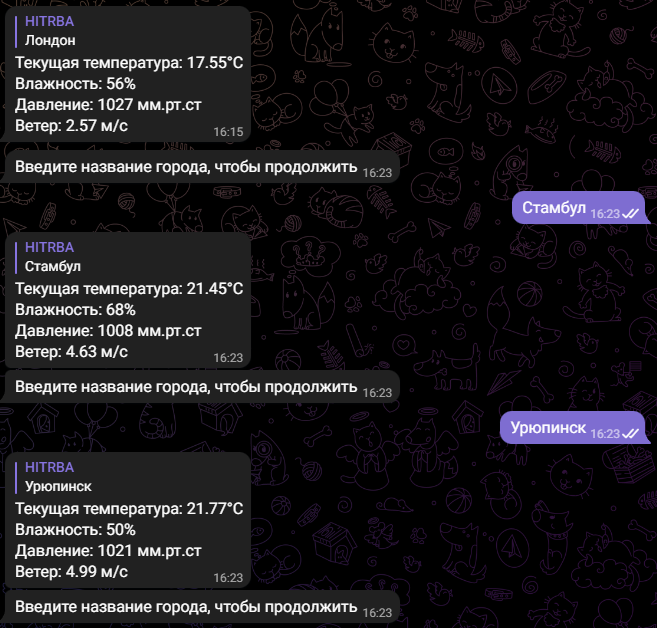


Пример работы:

Из. 2 – Первый пример работы



Из. 3 – Второй пример работы



Таким образом, тесты завершены. Ссылка на github с кодом и максимально коротким описанием:

<https://github.com/NoMoreSweet/BotAndreuha>

**Результаты и выводы**

В результате прохождения учебной практики были изучены технологии

разработки чат-бота в сети Telegram на языке Python, выполнены следующие задачи:

- Изучены, установлены и подключены нужные инструменты разработки;

- Настроена среда разработки;

- Разработан чат-бот со своим особенным функционалом;

- Выполнено тестирование чат-бота.

- В ходе практики был создан и настроен чат-бот в сети Telegram, помогающий в повседневной жизни любому желающему узнать погоду.