**Московский государственный технический**

**университет им. Н.Э. Баумана**

**Факультет ИУ**

**Кафедра ИУ5**

**Курс «Основы информатики»**

**Отчет лабораторной работе №5**

Выполнил студент группы ИУ5-36Б:

Турланов B. E.

Подпись и дата:

Проверил преподаватель каф.:

Нардид А. Н.

Подпись и дата:

Москва, 2024 г

**Разработка бота на основе конечного автомата для Telegram с использованием языка Python.**

**Цель лабораторной работы:** изучение разработки ботов в Telegram.

**Требования к отчету:**

Отчет по лабораторной работе должен содержать:

1. титульный лист;
2. описание задания;
3. текст программы;
4. экранные формы с примерами выполнения программы.

**Задание:**

1. Разработайте бота для Telegram. Бот должен реализовывать конечный автомат из трех состояний.

**tg\_bot.py**

import requests  
import datetime  
from config import tg\_bot\_token, open\_weather\_token  
import telebot  
  
bot = telebot.TeleBot(tg\_bot\_token)  
  
@bot.message\_handler(commands=["start"])  
def start\_command(message):  
 bot.reply\_to(message, "Привет! Напиши мне название города и я расскажу тебе о погоде!")  
  
@bot.message\_handler(func=lambda message: True)  
def get\_weather(message):  
 code\_to\_smile = {  
 "Clear": "Ясно \U00002600",  
 "Clouds": "Облачно \U00002601",  
 "Rain": "Дождь \U00002614",  
 "Drizzle": "Дождь \U00002614",  
 "Thunderstorm": "Гроза \U000026A1",  
 "Snow": "Снег \U0001F328",  
 "Mist": "Туман \U0001F32B"  
 }  
  
 try:  
 r = requests.get(  
 f"http://api.openweathermap.org/data/2.5/weather?q={message.text}&appid={open\_weather\_token}&units=metric"  
 )  
 data = r.json()  
  
 if data.get("cod") != 200:  
 bot.reply\_to(message, "\U00002620 Проверьте название города \U00002620")  
 return  
  
 city = data["name"]  
 cur\_weather = data["main"]["temp"]  
  
 weather\_description = data["weather"][0]["main"]  
 wd = code\_to\_smile.get(weather\_description, "Посмотри в окно, не пойму что там за погода!")  
  
 humidity = data["main"]["humidity"]  
 pressure = data["main"]["pressure"]  
 wind = data["wind"]["speed"]  
 sunrise\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunrise"])  
 sunset\_timestamp = datetime.datetime.fromtimestamp(data["sys"]["sunset"])  
 length\_of\_the\_day = sunset\_timestamp - sunrise\_timestamp  
  
 bot.reply\_to(message, f"\*\*\*{datetime.datetime.now().strftime('%Y-%m-%d %H:%M')}\*\*\*\n"  
 f"Погода в городе: {city}\nТемпература: {cur\_weather}C° {wd}\n"  
 f"Влажность: {humidity}%\nДавление: {pressure} мм.рт.ст\nВетер: {wind} м/с\n"  
 f"Восход солнца: {sunrise\_timestamp}\nЗакат солнца: {sunset\_timestamp}\n"  
 f"Продолжительность дня: {length\_of\_the\_day}\n"  
 f"\*\*\*Хорошего дня!\*\*\*"  
 )  
  
 except Exception as e:  
 bot.reply\_to(message, "U00002620 Произошла ошибка, попробуйте еще раз. U00002620")  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 bot.polling(none\_stop=True, timeout=30)



