

2. Офіційний сайт компанії Ammann Group. Режим доступу: info.aag@ammann-group.com.

3. Невлюдов І.Ш. Автоматизована система керування технологічними процесами в SCADA системі TRACE MODE 6: Навчальний посібник / І.Ш. Невлюдов, А.О. Андрусевич, В.В. Євсєєв, С.С. Максимова, М.Г. Стародубцев, В.В. Невлюдова. – Кривий Ріг: КК НАУ, 2018. – 316 с.

4. Суміші асфальтобетонні і асфальтобетон дорожній та аеродромний. Технічні умови: ДСТУ Б В.2.7 – 119:2011 -[Чинний від 01-10-2011] – К.: Мінрегіон України, 2011. – 59 с. (Національний стандарт України).

5. Офіційний сайт компанії ЧАО «Кредмаш». Режим доступу: <http://www.kredmash.com>.

ТЕЛЕГРАМ БОТ ПОМІЧНИК ПОДОРОЖУВАЛЬНИКА

Батура Михайло Віталійович

Харківський національний автомобільно-дорожній університет, Харків

Вступ

Сучасний світ характеризується стрімким розвитком інформаційних технологій, які значно впливають на різні аспекти нашого життя, включаючи комунікацію, доступ до інформації та виконання рутинних завдань. Однією з ключових тенденцій останніх років є зростання популярності месенджерів та чат-ботів, які надають можливість автоматизувати багато процесів та зробити взаємодію з технологіями більш зручною та ефективною. Telegram, як один з провідних месенджерів, пропонує розробникам потужний інтерфейс програмування додатків (API), що дозволяє створювати інтерактивних та функціональних ботів для вирішення широкого спектру задач.

Актуальність теми

З кожним роком збільшується кількість людей, які активно подорожують, працюють у міжнародних компаніях або просто цікавляться ситуацією в інших країнах. Глобалізація, розвиток міжнародних зв'язків та збільшення мобільності населення створюють нові виклики і потреби, серед яких особливе місце займає доступ до оперативної та достовірної інформації.

Подорожі, незалежно від їх мети, вимагають ретельної підготовки. Одним з найважливіших аспектів цієї підготовки є отримання актуальної інформації про погоду в місці призначення.

Вчасно отримані дані про погодні умови дозволяють мандрівникам краще планувати свої дії, вибирати відповідний одяг, розуміти можливі погодні ризики та забезпечувати свій комфорт та безпеку. Окрім цього, актуальна інформація про валютні курси є не менш важливою. Вона допомагає орієнтуватися в місцевій економічній ситуації, правильно розраховувати свої витрати, а також здійснювати вигідні обмінні операції.

Основна Частина

Архітектура розробленого Telegram-бота побудована на основі фреймворку `aiogram` для створення ботів мовою Python. Взаємодія з Telegram Bot API здійснюється через об'єкти класів **Bot** і **Dispatcher**, які забезпечують отримання та обробку повідомлень від користувачів.

Ключовим компонентом системи є механізм обробників подій, реалізований через декоратори бібліотеки `aiogram`. Для кожного типу вхідних повідомлень (команда, текст тощо) визначені відповідні обробники, які викликаються асинхронно при надходженні даних.

Основні функції бота включають:

1. **Обробка команди `/start`**: Ця команда ініціалізує бота та виводить вітальне повідомлення з інструкціями. Користувачу одразу пропонується головне меню з клавіатурою.

```
@bot.message_handler(commands=['start'])
def start(message):
    markup = types.ReplyKeyboardMarkup(row_width=2)
    btn1 = types.KeyboardButton('Погода')
    btn2 = types.KeyboardButton('Валюта')
    btn3 = types.KeyboardButton('Конвертація валют')
    markup.add(btn1, btn2, btn3)
    bot.send_message(message.chat.id, 'Привіт, обери що ти хочеш дізнатися:', reply_markup=markup)
```

Користувач отримує повідомлення з меню, яке містить кнопки "Погода", "Валюта" та "Конвертація валют".

2. **Обробка команди "Погода"**: Бот запитує у користувача назву міста та надає актуальну інформацію про погоду в цьому місті, використовуючи API `OpenWeatherMap`.

```
@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'погода':
        bot.send_message(message.chat.id, 'Введіть назву міста для отримання погоди:')
        bot.register_next_step_handler(message, get_weather)

def get_weather(message):
    city = message.text.strip()
    weather = get_weather_data(city)
    bot.send_message(message.chat.id, weather)

def get_weather_data(city):
    # Запит до API OpenWeatherMap
```

```

url = f"http://api.openweathermap.org/data/2.5/
weather?q={city}&appid={WEATHER_API}"
response = requests.get(url)
data = response.json()
if data['cod'] == 200:
    main = data['main']
    weather_desc =
data['weather'][0]['description']
    temp = main['temp'] - 273.15 # Конвертація з
Кельвіна в Цельсій
    return f"Погода в {city}:\nТемпература:
{temp:.2f}°C\nОпис: {weather_desc}"
else:
    return "Місто не знайдено, будь ласка,
перевірте введену назву."

```

3. Обробка команди "Валюта": Бот запитує у користувача назву міста та надає інформацію про валюту країни, в якій знаходиться це місто, використовуючи API Rest Countries.

```

@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'валюта':
        bot.send_message(message.chat.id, 'Введіть
назву міста для отримання інформації про валюту:')
        bot.register_next_step_handler(message,
get_currency)

def get_currency(message):
    city = message.text.strip()
    currency = get_currency_data(city)
    bot.send_message(message.chat.id, currency)

def get_currency_data(city):
    # Запит до API Rest Countries
    url =
f"https://restcountries.com/v3.1/capital/{city}"
    response = requests.get(url)
    data = response.json()
    if data:
        currency_info = data[0]['currencies']
        currency_name = list(currency_info.keys())[0]
        currency_symbol =
currency_info[currency_name]['symbol']

```

```

        return f"Валюта країни, де знаходиться
{city}:\nНазва:                {currency_name}\nСимвол:
{currency_symbol}"
    else:
        return "Місто не знайдено, будь ласка,
перевірте введене назву."

```

4. Обробка команди "Конвертація валют": Бот запитує у користувача суму та надає можливість вибору валютної пари для конвертації.

```

@bot.message_handler(content_types=['text'])
def handle_message(message):
    text = message.text.strip().lower()
    if text == 'конвертація валют':
        bot.send_message(message.chat.id, 'Введіть
суму для конвертації:')
        bot.register_next_step_handler(message,
get_amount)

def get_amount(message):
    global amount
    amount = float(message.text.strip())
    markup = types.ReplyKeyboardMarkup(row_width=2)
    btn1 = types.KeyboardButton('USD/EUR')
    btn2 = types.KeyboardButton('EUR/USD')
    btn3 = types.KeyboardButton('USD/GBP')
    btn4 = types.KeyboardButton('Друге значення')
    markup.add(btn1, btn2, btn3, btn4)
    bot.send_message(message.chat.id, 'Виберіть
валютну пару для конвертації:', reply_markup=markup)

@bot.message_handler(content_types=['text'])
def handle_conversion(message):
    text = message.text.strip().lower()
    if text in ['usd/eur', 'eur/usd', 'usd/gbp']:
        from_currency, to_currency = text.split('/')
        converted_amount = convert_currency(amount,
from_currency, to_currency)
        bot.send_message(message.chat.id,
f"Конвертована сума: {converted_amount:.2f}
{to_currency}")
    elif text == 'друге значення':
        bot.send_message(message.chat.id, 'Введіть
пару кодів валют (наприклад, USD/EUR):')
        bot.register_next_step_handler(message,
custom_conversion)

```

```

def custom_conversion(message):
    from_currency, to_currency =
message.text.strip().upper().split('/')
    converted_amount = convert_currency(amount,
from_currency, to_currency)
    bot.send_message(message.chat.id, f"Конвертована
сума: {converted_amount:.2f} {to_currency}")

def convert_currency(amount, from_currency,
to_currency):
    # Використання бібліотеки CurrencyConverter для
конвертації валют
    converted_amount = currency.convert(amount,
from_currency, to_currency)
    return converted_amount

```

Для реалізації користувацького інтерфейсу використовуються спеціальні клавіатури, створені за допомогою класу **ReplyKeyboardMarkup**. Головне меню містить кнопки для переходу до режимів отримання погоди, валютних курсів та конвертації валют.

Висновок

Розробка Telegram-бота є актуальною задачею в сучасному глобалізованому світі, де швидкий та точний доступ до інформації відіграє важливу роль. Створений бот забезпечує користувачам зручний інструмент для отримання актуальних даних про погоду, валютні курси та здійснення конвертації валют, використовуючи сучасні технології машинного перекладу та обробки природної мови.

У процесі роботи було реалізовано ключові функції бота, включаючи отримання даних про погоду, валютні курси та конвертацію валют. Використання фреймворку aiogram дозволило створити ефективний та стабільний бот з інтуїтивно зрозумілим інтерфейсом. Основними перевагами розробленого бота є його модульність, гнучкість та можливість подальшого розширення функціоналу. Застосування принципів об'єктно-орієнтованого програмування забезпечило зручність супроводу та розвитку проекту.

Список літератури

1. Googletrans: Free and Unlimited Google translate API for Python [Електронний ресурс]. Режим доступу: <https://py-googletrans.readthedocs.io/en/latest/>. Дата доступу: 15.04.2024
2. aiogram 3.7.0 documentation [Електронний ресурс]. Режим доступу: <https://docs.aiogram.dev/en/latest/>. Дата доступу: 15.04.2024
3. OpenWeatherMap API documentation [Електронний ресурс]. Режим доступу: <https://openweathermap.org/api>. Дата доступу: 15.04.2024
4. Rest Countries API documentation [Електронний ресурс]. Режим доступу: <https://restcountries.com>. Дата доступу: 15.04.2024