

УДК 004

МЕТОДОЛОГИЯ АСИНХРОННОГО ПРОГРАММИРОВАНИЯ НА ПРИМЕРЕ РЕАЛИЗАЦИИ TELEGRAM-БОТА «ПОДБОР ОДЕЖДЫ ПО ПОГОДНЫМ УСЛОВИЯМ»

Михайлов Д.Р.

федеральное государственное бюджетное образовательное учреждение высшего образования «Чувашский государственный университет имени И.Н. Ульянова»,
Российская Федерация, Чувашская Республика, г. Чебоксары

Научный руководитель: Ковалев С.В. – канд. техн. наук, доцент, доцент кафедры ВТ

Аннотация. В статье рассматривается разработка Telegram-бота с использованием метода асинхронного программирования для персональной рекомендации выбора одежды пользователям в зависимости от погодных условий.

Ключевые слова: асинхронное программирование, система управления базами данных, кроссплатформенное ПО.

Введение. В области разработки многопоточных программ зачастую возникают дискуссии выбора метода реализации. Всего существует три основных метода: многопоточная программа, многопроцессорная программа и асинхронная программа.

Многопоточная программа состоит из нескольких потоков внутри одного процесса, которые выполняются параллельно. Таким образом получается выигрыш во времени за счет распараллеливания вычислений, но возникает ряд проблем:

Возрастает сложность программы, затрачивается больше времени и трудовых ресурсов на проектирование, разработку и поддержку параллельного кода

В частности, в языке *Python* «настоящая» параллельность невозможна из-за *GIL (Global Interrupt Lock)*

Многопроцессорная программа состоит из нескольких процессов, каждый из которых обладает своими изолированными потоками. По сравнению с многопоточной программой, разработка многопроцессорной программы в разы увеличивает затраты на человеко-часы и ресурсы компьютера.

Асинхронная программа выполняется в одном потоке. При выполнении долгих процессорно-независимых операций, таких как: запрос к удаленному веб-серверу, запрос к базе данных; процессор «простаивает». Время простоя процессора можно использовать на выполнения другой операции. В этом и заключается принцип асинхронности [1].

Основная часть. Таким образом, наиболее оптимальным выбором метода реализации является именно асинхронный, за счет его простоты проектирования и использования.

API Telegram предоставляет широкие возможности для разработки, одновременно являясь кроссплатформенным средством. Приложение, разработанное на данной платформе, доступно как на мобильных устройствах (*iOS, Android* и прочее), так и на десктопных.

Главным инструментом разработки стал язык программирования высокого уровня *Python 3.7*. Данный язык поддерживает библиотеки *aiogram* и *asyncio* для асинхронной реализации приложения. Ключевое преимущество асинхронного подхода в создании Telegram-бота – это отсутствие блокировки программой основного процесса при выполнении таких долгих операций, как *HTTP*-запрос или запрос к базе данных. Во время ожидания ответа на запрос, основная программа может продолжить обработку других команд, что критически важно при проектировании приложения, которым будут пользоваться большое количество пользователей [2].



Рис. 1. Схема работы асинхронной программы

При использовании библиотеки *aiogram* на этапе проектирования приложения необходимо учитывать, что все сопутствующие используемые библиотеки должны поддерживать асинхронность. Так, например, стандартная библиотека *sqlite3* не поддерживает асинхронный ввод-вывод. Поэтому необходимо использовать ее улучшенный вариант – *aiosqlite*, поддерживающий требуемые функции. Данная библиотека не делает базу данных асинхронной, а лишь автоматически помещает запросы в очередь, не давая одновременным запросам конфликтовать между собой.

Все необходимые объекты и методы для асинхронного программирования хранятся в библиотеке *asuncio*. Главные особенности этой библиотеки: она позволяет создавать планирование выполнения асинхронных функций – корутин. Корутина – это сопрограмма, от которой ожидают, что она будет отдавать управление обратно в цикл событий. Отличие корутины от обычной функции в коде задается ключевым словом *asunc* при ее объявлении. Асинхронная функция вызывается с помощью ключевого слова *await* исключительно либо в планировщике выполнения сопрограмм, либо также в другой асинхронной функции.

Основная функция *Telegram*-бота – обработка команд и сообщений от пользователя. Для принятия и обработки сигналов от *API Telegram* в библиотеке *aiogram* предусмотрен класс *Dispatcher*. Объект этого класса может существовать в единственном экземпляре с указанным токеном бота. Это ограничение существует для того, чтобы только одна работающая программа могла принимать и обрабатывать сигналы. В противном случае, две или более программ конфликтовали бы между собой.

Для обработки команд и сообщений от пользователя необходимо переопределить метод *message_handler* класса *Dispatcher* с помощью декоратора. Декораторы позволяют модифицировать выполняемую функцию, значения ее параметров и ее результат без изменения исходного кода этой функции [3].

`@dispatcher.message_handler(commands=['команда1', 'команда2, ...])` – такой вид имеет декоратор при переопределении функции обработки команд. Для обработки текстовых сообщений в параметре декоратора необходимо указать *content_types=['text']*.

Формирование рекомендаций основано на погоде в ближайшие 6 часов. Парсинг данных о погоде осуществляется с сервиса *openweathermap.org* – популярного сервиса, предоставляющего API для получения различной информации о погоде.

`requests.get("http://api.openweathermap.org/data/2.5/weather", params={'id': city_id, 'units': 'metric', 'lang': 'ru', 'APPID': WEATHER_API_TOKEN})` – пример запроса на сервер сервиса погоды, где *city_id* – идентификатор города, *WEATHER_API_TOKEN* – личный токен доступа.

Рекомендательная система в текущем ее виде реализована в виде экспертной системы: отправляется запрос к базе данных на получение списка предметов одежды, подходящие в данную погоду. Из этого списка берется одна случайная вещь для каждого уровня одежды (головной убор, торс, низ).

`SELECT name FROM Clothes WHERE (gender=''' + str(gender) OR gender='Унисекс') AND + str(temp) + >= min AND + str(temp) + <= max AND type= + str(i) + ORDER BY RANDOM() LIMIT 1`

Такой SQL-запрос используется для получения необходимого списка одежды.

Система уведомлений реализована с помощью библиотеки `aioschedule`. Данная библиотека позволяет запланировать выполнение какой-либо асинхронной функции в заданное время с заданным периодом. Так как необходимо присылать рекомендации раз в сутки, применяется следующая команда:

```
aioschedule.every().day.at(notifTime).do(getRecommend, userID, userData)
```

где `notifTime` – время уведомления, `getRecommend()` – функция для формирования рекомендации, `userID` и `userData` – необходимая информация о пользователе.

Для того, чтобы данная команда не блокировала основной поток, необходимо создать асинхронную функцию, которая раз в секунду проверяет, есть ли уведомления, которые необходимо отправить:

```
async def scheduler():
    while True:
        await aioschedule.run_pending()
        await asyncio.sleep(1)
```

Для того, чтобы данная функция работала асинхронно с момента старта выполнения программы, при запуске мы создаем соответствующее событие:

```
asyncio.create_task(scheduler())
```

Заключение. Таким образом, метод асинхронного программирования является одним из наиболее оптимальных решений при проектировании телеграм-бота.

Список литературы

1. Caleb Hattingh, *Using Asyncio in Python: Understanding Python's Asynchronous Programming Features*
2. Документация по языку программирования Python 3.7 [Электронный ресурс]. – Режим доступа: <https://docs.python.org/3.7/>
3. Документация Telegram Bot API [Электронный ресурс]. – Режим доступа: <https://core.telegram.org/bots/api>

UDC 004

ASYNCHRONOUS PROGRAMMING METHODOLOGY ON THE EXAMPLE OF THE TELEGRAM BOT IMPLEMENTATION "SELECTION OF CLOTHES ACCORDING TO WEATHER CONDITIONS"

Mikhailov D.R.

Federal State Budgetary Educational Institution of Higher Education "I.N. Ulyanov Chuvash State University", Russian Federation, Chuvash Republic, Cheboksary

Kovalev S. – PhD, associate professor, associate professor of the Department of VT

Annotation. The article discusses the development of a Telegram bot using the asynchronous programming method to personally recommend the choice of clothes to users depending on weather conditions.

Keywords: asynchronous programming, database management system, cross-platform software.