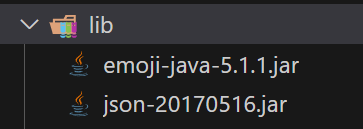
**Lesson 26. EmojiBot**

Після знайомства із тим, як почати працювати із телеграм-ботом, пропонуємо навчитись додавати у повідомлення бота емоджі, які давно вже стали невд’ємною частиною спілкування онлайн.

Емоджі дозволять відповіді бота зробити «живішими» і приємнішими для користувача.

**Emoji**

Для використання емоджі у нашому боті знадобляться ще додаткові бібліотеки:

Далі на прикладі простого бота з’ясуємо як додати до повідомлення емоджі. Перш ніж це зробити варто заглянути в клас Emoji з відповідної бібліотеки, аби з’ясувати з яких з чого складається об’єкт емоджі.

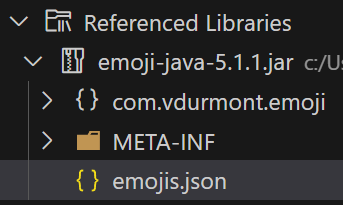
Нас найбільше будуть цікавити ось ці три властивості:

**aliases** – «псеводніми» емоджі, щось наче кодові рядки, за якими емоджі можна знайти;

**tags** – за тегами емоджі можуть бути роздідлені на категорії, або тег може містити якусь допоміжну інформацію про емоджі;

**unicode** – код символу емоджі (саме він повинен бути в тексті повідомлення, що телеграм відобразив його як картинку);

Детальну інформацію про емоджі можна знайти у файлі із бібліотеки emoji-java у наступному вигляді:





Також з’ясувати еліас емоджі можемо в самому Telegram навівши на якийсь із них курсором. Еліас зазвичай починається із закінчується символом «**:**».

Знаючи еліас можемо відтворити юнікод емоджі у повідомленні. Для цього можна скористатись одним із двох запропонованих способів:

1. Клас **EmojiParser**;

Потрібний нам метод **parseToUnicode**() приймає у якості параметру рядок тексту, а повертає такий самий рядок, але всі знайдені еліаси в ньому заміщує юнікодами емоджі, якщо такі є.

1. Клас **EmojManager**;

Даний клас дозволяє за допомогою методу **getForAlias**() знайти об’єкт емоджі за його еліасом. Далі з отриманого об’єкту можна отримати про нього (про емоджі) всі необхідні дані, втому числі юнікод, який потім додати у текстовий рядок повідомлення.

У якості прикладу та поля для досліджень пропонуємо відтворити приклад на скріншоті нижче. В рамках телеграм бота можна поекпериментувати із еліасами, емоджі та запропонованими методами:

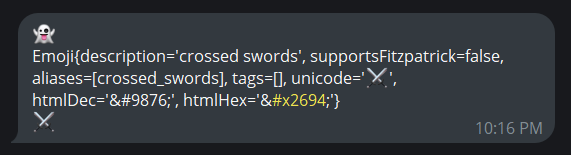


Дані класи (EmojiParser та EmojiManager) мають ще багато іншого функціоналу, який ми в рамках курсу не будемо використовувати. Однак це не значить, що самостійно в нього заглибитись ви (учні) не можете за допомогою функціоналу автозаповнення та перегляду файлів бібліотек.

**Practice Tasks**

**Завдання 1**. Створіть телеграм бота, використавши скріншот із розділу вище. Скористайтесь шаблоном (проєкт templatebot) у якості відправної точки. Бот повинен у відповідь на будь-яке текстове повідомлення надсилати користувачеві повідомлення як показано на скріншоті на наступній сторінці.

Спробуйте також змінити емоджі на власні – тобто які виберете самостійно у телеграм, або з файлу бібіліотеки.



**Завдання 2**. Самостійно створіть бота із шаблону, або змініть попереднього, який буде у відповідь на ваше повідомлення, обирати із надсилати емоджі, еліас якого відповідає отриманому повідомленню. Наприклад на слово alian (Alian або aLiEn) у відповідь надсилати емоджі з прибульцем.

Загалом перевірити правильність роботи бота можна за скріншотом нижче:

**Завдання 3**. Самостійно створіть бота із шаблону, або змініть попереднього

**Завдання 1.** Повіторіть вдома код найпростішого телеграм-бота, або ж Mirror Bot’а. Переконайтесь, що все працює правильно.

**Homework**