

API та обгортка. За допомогою Slack WEB API було отримано json файл із історією повідомлень, який був перетворений у словник за допомогою Python обгортки SlackClient. За допомогою Slack RTM API відбувалось зчитування щойно надісланих повідомлень та пост нових повідомлень у канал знову ж таки з використанням Python-обгортки. Тому потрібно відзначити потужність Python-обгортки SlackClient, яка виконувала усі процеси пов'язані із парсингом даних. Здобутий досвід: розуміння функціоналу та призначення API для роботи з певним ресурсом, аналіз документації обгортки для полегшення роботи з API, використання функціоналу обгортки для реалізації необхідних процесів.

Передача даних по мережі. Здобуто базове розуміння та призначення json та xml файлів, хоча безпосередня робота з ними була повністю у функціоналі Python-обгортки.

Абстрактний тип даних. Здобутий досвід створення(проектування та реалізації) власних абстрактних типів даних, поєднання їх із базовими структурами даних(масив та динамічний масив), його використання у програмі. Підвищилось розуміння принципів ООП, адже абстрактні типи даних були представлені у вигляді класу. Крім цього були створені окремі класи для роботи із даними та для використання їх у абстрактних типах даних. Взагалі у проекті було поєднано два принципи програмування: об'єктно-орієнтований та функціональний.

Сторонні бібліотеки. Однією з найважливіших складових проекту була пайтон бібліотека для gensim для лінгвістичного аналізу документів, рядків, слів. Вона застосовувалася для визначення пошуку семантично однакових запитань. Здобуто досвід роботи із документацією повністю невідомої бібліотеки та її адаптація до проекту. Набута базова техніка семантичного аналізу документів на основі TF-IDF та LSI моделей.

Автоматична генерація документації. Здобуто досвід автоматичної генерації документації за допомогою sphinx у вигляді html-сторінки: створення *.rst файлів для генерації відповідних html документів, налаштування конфігурації документації(conf.py).

Створення дистрибутиву програми. Здобуто навички побудови дистрибутиву проекту для платформ Linux та Windows за допомогою вбудованої у Python кросплатформної утиліти distutils: створення та налаштування файлу конфігурацій(MANIFEST.in), налаштування файлу установки(setup.py), встановлення програми із дистрибутиву.

Застосування створеного проекту. На даному етапі створений бот здатний працювати із Slack каналами створюючи базу запитань-відповідей із нуля та з каналом із предмету Основи програмування із уже існуючою початковою базою.

Перспективи розвитку.

- універсалізація бота для роботи у різних середовищах(не тільки у Slack)
- створення алгоритму виділення запитань та відповідей у базі повідомлень(у зв'язку із незначною кількістю повідомлень вибиралися вручну)
- створення бази даних запитань та відповідей(наразі текстові файли)
- покращення алгоритму визначення типу речення(запитання, відповідь звичайне) з особливим наголосом на запитанні
- покращення алгоритму визначення найбільш подібного запитання, зокрема врахування синонімічних рядів(синонімів конкретного слова)

- додавання кнопки "Відповісти на запитання" для уникнення постійного використання звертань(@username). Додавання можливості вибору запитання для відповіді(на подібні випадуючого списку)
- реліз бота як Slack App для встановлення у інших командах