

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут» імені Ігоря Сікорського
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Лабораторна робота № 4

З дисципліни: «Компоненти програмної інженерії – 2.
Моделювання та аналіз програмного забезпечення»

Виконав:

студент групи ІТ-73

Старовойтов Руслан Олександрович

Перевірив:

Галушко Д. О.

Київ 2020

Тема: Розробка конструктора Telegram-ботів для замовлень.

Мета: Створення базових інтерфейсів для сервера ботів.

Перед розробкою сервера ботів необхідно створити інтерфейси. Найбільш головними інтерфейсами є `ISession`, `ITreeNode`, `IMarkupTree` та `IExtendedBot`. Основна ідея проста: бот містить у собі дерево розмітки, що складається з вузлів. Також в боті знаходяться сесії користувачів, які можуть переміщуватися по вузлах дерева.

Оскільки початково планувалася можливість використання декількох мов у боті, то були додані інтерфейси `ITranslatable`, `IMetaMessage`, `IMetaReplyMarkup`, `ITextMessagesManager` та `ITranslator`. Інтерфейси з префіксом «Meta» зберігають не сам текст, який необхідно відправити користувачу, а ключі, які за допомогою інтерфейсу `ITranslator` перекладаються на обрану користувачем мову.

Щоб була можливість створювати замовлення, додавати товари у корзину та зберігати список усіх товарів і взагалі різні додаткові змінні (як, наприклад, інформацію про адрес чи час доставки), був створений інтерфейс `IVariablesContainer`, який дозволяє працювати зі змінними будь-якого типу.

Інтерфейс `IBot` може використовуватися у випадку, якщо необхідно запустити власноруч створеного (не через веб-сайт) бота на сервері ботів. Такі боти, наприклад, можуть бути створеними на замовлення і не мати звичних вузлів, дерева розмітки, сесій чи статистики.

Інтерфейс `IGlobalFilter` призначений для обробки спеціальних повідомлень з будь-якого місця дерева. Такими повідомленнями можуть бути, наприклад, команди на демонстрацію вмісту корзини.

Крім названих вище інтерфейсів ще існують й інші, які можна побачити на діаграмі нижче. Призначення багатьох з них доволі очевидне і можна зрозуміти з назви. Інші ж були створені, щоб існуючі інтерфейси не ставали занадто великими та для відповідності Interface Segregation Principle.

