

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 № 5

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

Виконав:

студент групи ІТ-73

Похиленко Олександр Андрійович

Перевірив:

Галушко Д. О.

Київ 2020

Тема: Розробка конструктора Telegram-ботів для замовлень.

Мета: Обрати спосіб взаємодії бота з користувачами та виділити основні типи вузлів дерева розмітки для сервера ботів.

Конструювання ботів виконується за рахунок створення дерева розмітки (markup tree) зі спеціальних вузлів (tree nodes). Вони можуть бути різних типів і виконувати різні функції. Щоб не заплутати користувача, на перших етапах необхідно спробувати обрати лише невелику кількість основних типів вузлів, які будуть найбільш необхідними.

Щоб зрозуміти, які ж саме типи вузлів потрібно обрати, спочатку необхідно визначити, як буде користувач взаємодіяти з ботом. Можна виділити такі основні способи взаємодії з ботами: з використанням природної мови (natural language processing: користувач пише повідомлення, як якщо б він розмовляв зі справжньою людиною), з використанням команд (починаються зі «/», наприклад, «/start») та з обиранням одного з можливих варіантів зі списку (наприклад, користувачу необхідно натиснути на одну з декількох inline- чи reply-кнопок).

Обробка природної мови є дуже важким завданням, тож відразу можна від нього відмовитися: такий конструктор ботів буде складно і реалізувати, і використовувати. Використання команд змушує людину запам'ятовувати їх чи писати кожен раз їхній перелік у повідомленні від бота – тоді цей спосіб майже не відрізняється від третього, але є менш зручним. Кнопки ж дозволяють дуже зручно реалізовувати навігацію, якщо писати на них коротке змістовне повідомлення. Наприклад, в боті для замовлень на кнопках можна писати назву категорій та товарів. При натисканні на кнопку повинно відправлятися нове повідомлення з новими кнопками. Якщо, наприклад, користувач натискає на кнопку з назвою товару, буде інтуїтивно зрозумілим, що бот відправить повідомлення з описом та ціною і кнопкою «Додати у корзину». Або, наприклад, що натиснувши на кнопку «Контактні дані» буде відправлено повідомлення з електронною поштою чи телефонним номером.

Отже, взаємодія користувача з ботом буде відбуватися через кнопки. Як можна було побачити з прикладів, кожен вузол повинен мати назву, яка буде відображатися на кнопці, та повідомлення. Крім тексту, у будь-якому повідомленні можуть ще бути прикріплені інші файли, наприклад, фото, відео, аудіо чи документ. Таким чином вже можна виділити найпростіший тип вузлів – **інформаційний**. Він має назву, текст повідомлення і може мати прикріплений файл. Крім цього, він має посилання на батьківський вузол та список дитячих вузлів, щоб можна було відображати їхні назви на клавіатурі для навігації.

Але можливі такі ситуації, коли вузол буде мати велику кількість дитячих вузлів, через що їх буде дуже складно відобразити на клавіатурі. Для вирішення цієї проблеми був введений ще один тип вузлів – **розділ**. Цей вузол має два підтипи: *блочний* та *сторінковий*. Блочний вузол розділу відправляє попередній перегляд дитячих вузлів (опис з однією inline-кнопкою «Детальніше») по 2 вузли у блоці і дозволяє виконувати навігацію герлу-кнопками «Назад» та «Вперед». Сторінковий вузол розділу відправляє inline-клавіатуру, що містить 6 кнопок з назвами дитячих вузлів та inline-кнопки для «перегортання» цих сторінок, при натисканні на які виконується оновлення клавіатури. Такі вузли розділу можна використовувати, наприклад, для меню.

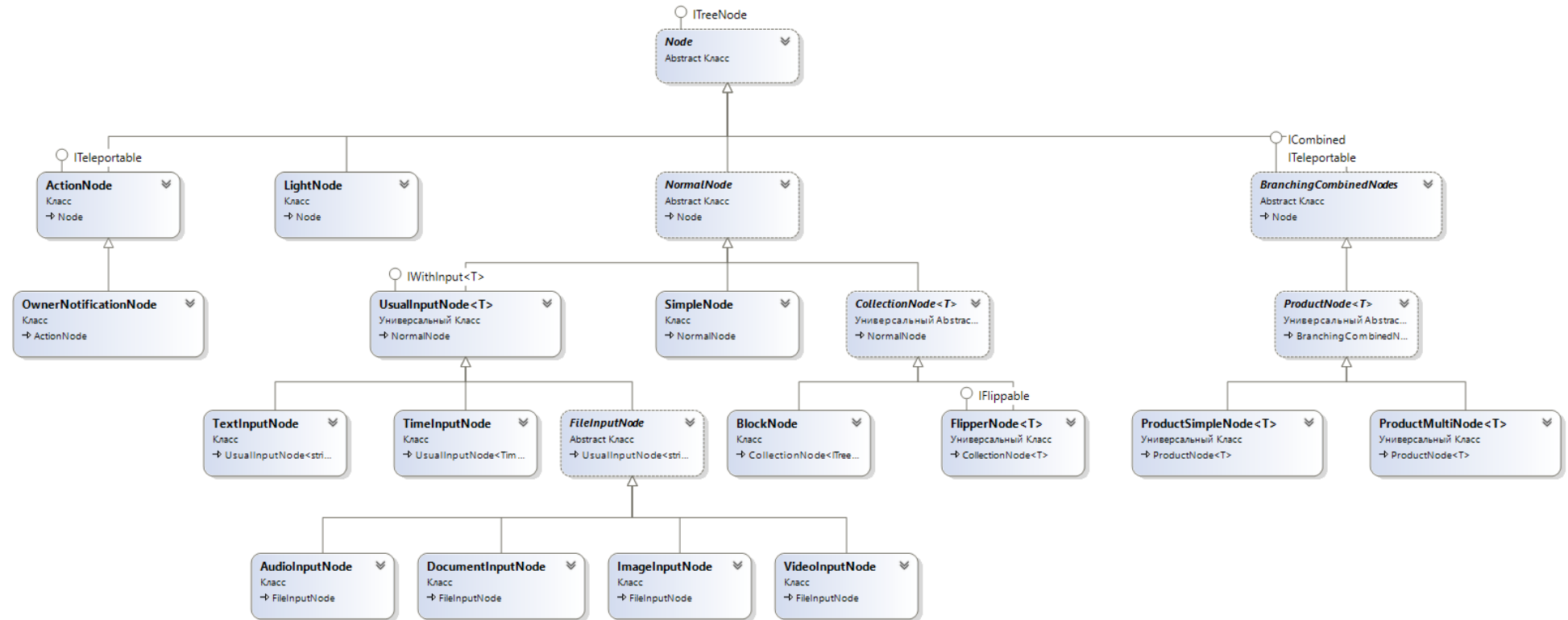
Як вже було вказано у прикладі, бота можна використовувати для роботи з товарами. Зрозуміло, що для надання користувачу можливостей додавання товарів у корзину, обрахування сумарної ціни та інше необхідно створити спеціальний тип вузлів – **товар**. Зазвичай багато товарів мають різні підтипи, а кожен підтип має свою ціну. Отже, було вирішено додати можливість створювати у кожному товарі певні підтипи та підвиди – «секції» та «підсекції». Для піци секціями можуть бути «діаметр» та «тип борта», а підсекціями цих секцій: «20 см», «30 см», «40 см» та «звичайний», «хот-дог», «з сиром» відповідно. При цьому надається можливість вказати для кожної секції повідомлення з описом, а для кожної комбінації (у даному прикладі їх

б) вказати відмінну ціну. При цьому товарний вузол також має 2 підтипи: *у вигляді звичайних вузлів* та *у вигляді мультивузла*. Товарний вузол у вигляді звичайних вузлів перетворюється на невелике дерево, де на кожному етапі у користувача запитують, яку підсекцію з наступної секції він хоче обрати. Тобто, у випадку з піцою спочатку питають, який діаметр бажає користувач обрати, а потім – який тип борта, після чого товар буде доданий у корзину. Мультивузол же дозволяє обрати підсекції кнопками – для кожної секції на клавіатурі створюється рядок з трьох кнопок – «Назад», кнопка з поточним значенням підсекції та «Вперед». Якщо ж секцій більше 6, то, як і у випадку зі сторінковим вузлом розділу, створюється ще один рядок з 2 кнопок – «Назад» і «Вперед» для перегортання сторінок. У самому низу знаходиться кнопка «Додати», що додає товар зі вказаними підсекціями у корзину.

Наступний тип вузлів – вузол **вводу даних**. Якщо повернутися до прикладу із замовленням піци, то цей вузол необхідний для того, щоб користувач міг вказати адресу, куди кур'єр повинен привести піцу. Цей тип взагалі має 6 підтипів: введення та відправка *тексту*, *часу* (у форматі HH:MM), *зображень*, *аудіо*, *відео* та *документів*. Це, наприклад, дозволяє вказати час, коли замовлення повинно бути доставлено. Відправка файлів може знадобитися у більш специфічних випадках, наприклад, документів – у принтцентрах.

Останній тип вузлів, який може додати користувач – вузол **відправки замовлення**. З точки зору користувача він дуже схожий на звичайний інформаційний вузол, але при переході на нього формується замовлення і все, що знаходиться у корзині (а також введені через спеціальні вузли дані) відправляється власнику бота. Власник бота ще може вказати, яку ж саме групу статусів він буде використовувати для швидких відповідей на замовлення через цей вузол. Так для замовлення піци може використовуватися, наприклад, якась група статусів «Для замовлень», де знаходяться статуси «прийнято», «у дорозі», «відмінено», «в обробці».

Спрощена діаграма класів основних вузлів, що використовуються у конструкторі ботів



З діаграми, яка наведена вище, можна побачити, що усі перераховані раніше типи вузлів разом з підтипами перетворилися на 21 клас. Інформаційний вузол – це SimpleNode. Вузли розділу – BlockNode та FlipperNode (від «flip» – «перегортати», оскільки він дозволяє перегортати сторінки). Товарні вузли – нащадки ProductNode. Вузли вводу даних – UsualInputNode. Відправки замовлень – OwnerNotificationNode.

Але все ж деякі класи потребують пояснення. ActionNode потрібен, оскільки у майбутньому можуть бути додані ще деякі вузли, при переході на який виконується дія. Це може бути зміна поточної мови, очистка кошика тощо. LightNode є вузлом попереднього перегляду («легким»), який використовується у блоковому вузлі розділу. Відмінність його від звичайних вузлів у тому, що він не використовує reply-клавіатуру, а має лише одну inline-кнопку. BranchingCombinedNodes – спеціальний клас для складених вузлів, коли один вузол реагує як декілька. Насправді це лише зручна обгортка, яка дозволяє працювати з цілою частиною дерева розмітки як з одним вузлом.

Крім цього, були додані ще й деякі нові інтерфейси, які не були розглянуті в минулій лабораторній. Інтерфейс ITeleportable свідчить про те, що замість списку дитячих вузлів цей вузол має лише одну вихідну точку і при переході на цей вузол відразу ж виконується перехід на наступний (тобто як якщо користувач відразу ж натиснув на кнопку переходу). Так, наприклад, вузол відправки замовлення відразу ж після відпрацювання «телепортує» користувача в корінь дерева. Через наявність таких порталів, насправді, розмітка перестає буди деревом, а стає сіткою, але назва залишилась, оскільки сама розмітка відображається в редакторі як дерево, а «портальні вузли» є більш виключенням з правил, ніж правилом. Інтерфейс IFlippable вказує на те, що вузол може мати декілька сторінок. IWithInput – що вузол приймає та зберігає дані. Вузли з інтерфейсом ICombined трохи по іншому оброблюються при додаванні у дерево розмітки, оскільки вони містять у собі ще вузли.