

Для своєї курсової роботи я створив UsersLikers ADT, який знаходиться у users_likers_adt.py. Цей абстрактний клас даних допомагає користувачу діставати інформацію про конкретні фото. Завдяки ньому значно спрощується робота програми RealFriends у файлі nicetwo.py. Для реалізації цього класу я також використав клас Node із файлу node.py, щоб вдосконалити роботу UsersLikers. Мета, яку виконує цей клас полягає у тому що за нумерацією фото від найновішої(позначається 1) до найстаршої(можна позначити як -1) ми можемо вибірково додати в абстрактний клас даних ті, які нам потрібні і пізніше діставати з них інформацію як і з кожної індивідуально, так і загальну по всіх одночасно.

UsersLikers містить 6 методів серед яких `__init__(self, user, api)`, `__str__(self)`, захищений метод `_get_userid(self, user)`, та публічні методи `add(self, v1=0, v2=0)`, `all_added(self)`, `media_likers(self, numb)`. Для роботи цього абстрактного типу даних імпортовано класи Node із node.py та клас InstagramAPI із файлу `__init__.py` папки InstagramAPI. А також імпортовано simplejson та urllib2 для роботи з інтернет даними у методі `_get_userid`.

Метод `__init__(self, user, api)` ініціалізує змінні для подальшого використання в ADT. User – це стрінгове значення в якому ми задаємо ім'я користувача фото з сторінки якого ми будемо використовувати. self.user одразу присвоюється значення результату функції `_get_userid(user)`. api це клас InstagramAPI із аргументами login та password щоб залогуватись для подальшої змоги використовувати дані API Instagram. Також в `__init__` створюється змінна list_of_added в яку записуватимуться результати виконання функції add для зручнішої роботи.

Метод add додає зображення до вузла за допомогою класу Node. Перше зображення і дані про нього збережуться як словник у self._head, і всі наступні за допомогою використання self._head.next. add має 2 аргументи v1 та v2 – порядкові номери картинок. Якщо запустити add без аргументів буде додано всі зображення користувача, якщо задати лише v1 то буде додано лише це конкретне фото під номером v1, якщо задати лише v2 то буде додано усі зображення від першого(найновішого) до того, під номером v2, і якщо ж задати обидва аргументи, то в межах між цими зображеннями всі фото будуть додані(включно із v1 та v2).

Метод `_get_userid` створений для того щоб перетворювати нікнейм користувача у user ID код. Для цього я використовую бібліотеку urllib2, яка дістає дані з веб мережі і отримує цей ID і багато іншої непотрібної інформації, тож потім із допомогою бібліотеки simplejson я перетворюю отримані дані у python dictionary і дістаю потрібний мені код, який і повертає цей метод.

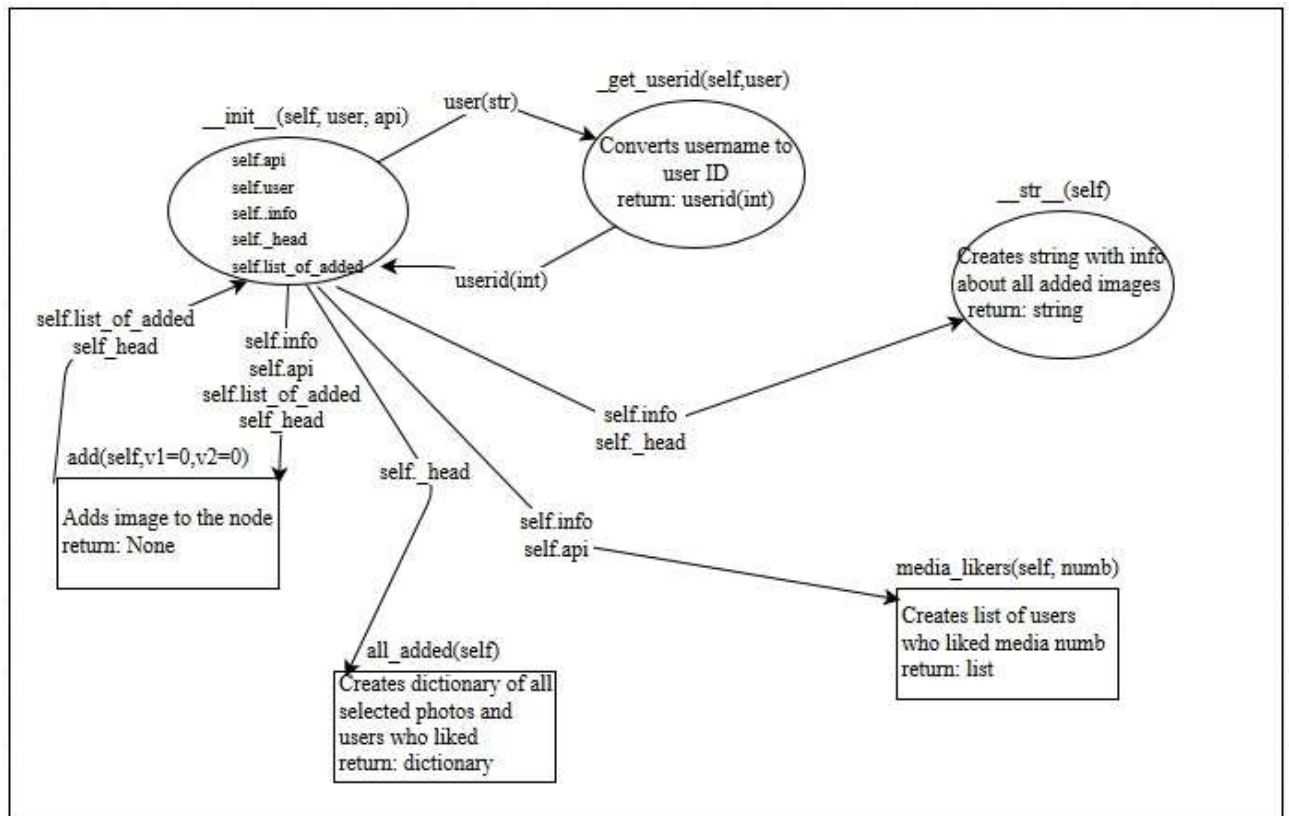
Метод all_added повертає словник із усіма доданими фото як ключами та користувачами які лайкнули це фото як значеннями. Для цього ми пробігаємось по всьому існуючому вузлі і по одному додаємо нові ключі у словник.

Метод media_likers повертає список користувачів, які лайкнули вибране фото. Номер фото інформація про яке береться задається в аргументі numb. Ця нумерація знову іде від новішого до найстаршого, які відповідно позначаються 1 та -1. Список складається із стрінгів, які є логінами користувачів.

Метод `__str__` повертає стрінг у який показує усі додані фото. `__str__` пробігається по вузлу починаючи від self._head і додає посилання на фото та кількість користувачів які це фото лайкнули. Кожне наступне фото принтиться із нового рядка і через двокрапку написано скільки відомих юзерів його лайкнуло. Проте якщо лайк стоїть від невідомого користувача, то він не враховуватиметься.

Цей ADT в основному використовується в програмі RealFriends, щоб складати статистику із вибраних фотографій по користувачах і кількості лайкнутих фотографій ними.

UsersLikers ADT



Структура даних Node

