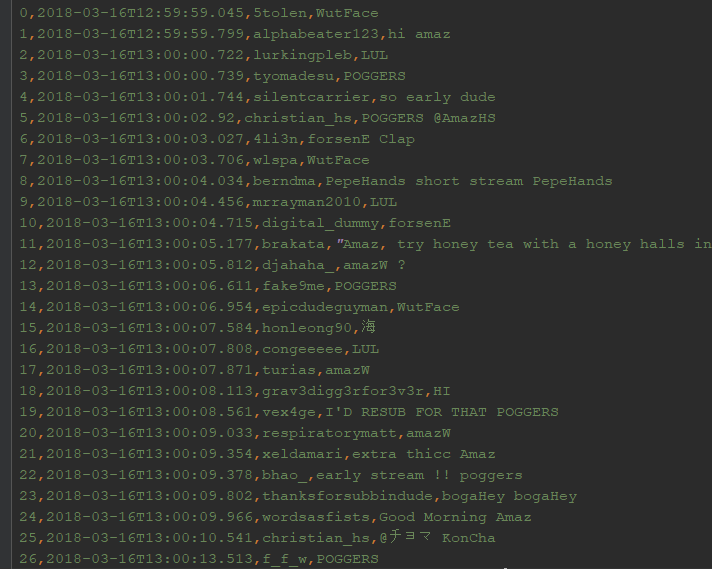
Опис абстрактного типу даних, який необхідний для вирішення задачі дослідження

Оскільки я працюю з чатом, де дуже важливі часові рамки(бо частина дослідження пов’язана з цим), то мені знадобиться абстрактний тип даних, який допоможе мені представити час і дату в зручному форматі - клас Date.



Як видно з скріншоту справа, час повідомлення мені повертається у формі year-mon-day’T’hour:min:sec.

Було б дуже зручно зробити клас, який допоможе представити цю дату у вигляді звичайної, а також брати для порівняння у дослідженні дати повідомлень.

Тому я реалізую клас Date, який допоможе мені спростити виконання завдання. У цьому класі будуть такі методи(основні, при потребі – можна збільшити к-сть методів):

* У методі \_\_init\_\_ перевіряється, чи були передані правильні дані у правильному форматі(у моєму випадку, формат буде year-month-day’T’hour:min:sec. Далі розраховується день за юліанським календарем(щоб потім було зручно порівнювати між собою різні дати)
* Далі кілька схожих методів:
  + Метод day повертає день
  + Метод year повертає рік
  + Метод month повертає місяць
  + Метод hour повертає годину
  + Метод minutes повертає хвивину
  + Метод seconds повертає секунди
* Метод in\_minutes повертає час у хвилинах без врахування дати
* Приватний метод \_is\_valid\_gregorian перевіряє, чи є валідною дата
* Метод \_is\_valid\_time перевіряє, чи є валідним час
* Метод \_\_str\_\_ повертає дату у зручному для користувача форматі
* Метод \_\_repr\_\_ повертає дату у зручному для розробника форматі
* Метод \_\_eq\_\_ перевіряє чи одна дата і час дорівнюють іншій
* Метод \_\_lt\_\_ порівнює, чи перша дата-аргумент була швидше, ніж друга
* Метод \_\_le\_\_ порівнює, чи перша дата-аргумент є меншою-рівною за другу дату-аргумент
* Приватний метод \_to\_gregorian переводить дату з юліанської системи в грегоріанську

Потім результати метода \_\_str\_\_ будуть записуватись разом з іншою інформацією(текст повідомлення, автор повідомлення) у файл з розширенням .csv.