

# АБСТРАКТНИЙ ТИП ДАНИХ

## ОПИС

Для реалізації проекту було вирішено розробити абстрактний тип даних Judgment для представлення судового рішення. Для того, щоб зберігати окремі рішення таким чином, потрібно спершу дістатися до них, прочитавши json-файл та проітерувавшись по знайдених рішеннях, тому також варто було б розробити для цього JudgmentsCollector.

### 1. Judgments Collector ADT

Читач для json-файлу отриманого в результаті запиту на колекцію рішень до судового API.

Якщо ми формуємо пошуковий запит для знаходження судових рішень, переважно їх знаходиться більше 1000, а API дозволяє виводити максимум 1000 рішень за один раз. Для того, щоб доступитися до решти рішень, потрібно надсилати запит ще раз, змінивши в ньому параметр start – зміщення у пошуку (дивись [опис судового API у папці DOCS](#)). Для зручності планується реалізувати пошук і обробку ВСІХ знайдених рішень за допомогою JudgmentsCollector.

JudgmentsCollector ініціалізовуватиметься приймаючи параметри API-запиту, формуватиме запит для отримання перших 1000 рішень, а потім переформовуватиме його на кожні наступні 1000 і так для отримання всіх рішень за даним запитом.

- JudgmentsCollector(text, number, justice, start, limit, judgment, adjudication\_date\_start, adjudication\_date\_end, distinct, sort): Створює екземпляр колекціонера даних за конкретним запитом
- Прихований атрибут api\_key: ключ доступу до API
- find\_judgements(): знаходить всі судові рішення за запитом і зберігає словник як атрибут
- judgements(): повертає словник із судовими рішеннями (судові рішення тут мається на увазі інформація про них, що повертається на запит колекції рішень, а вона без самого тексту рішень – дивись документацію до API) або None якщо їх немає.
- ids(): повертає номери документів судових рішень - список
- courts(): повертає список судів(код суду), що приймали судові рішення
- judges(): повертає список суддів(за ПІБ), що приймали рішення
- cases(): повертає список проваджень, в рамках яких були ці судові рішення

## 2. Judgement ADT

Коли ми знайдемо всі необхідні судові рішення нам потрібно буде їх належним чином зберігати.

Для презентації судового рішення буде розроблено тип даних Judgement.

- Judgement(category\_code, adjudication\_date, justice\_kind, cause\_num, doc\_id, judgment\_code, receipt\_date, judge, text)
- judge(): повертає суддю, що приймав рішення
- date(): повертає дату прийняття рішення
- keywords(): повертає список ключових слів у рішенні
- includes(word): перевіряє рішення на наявність у ньому певного слова та повертає булеве значення
- codes(): повертає список кодексів використаних у рішенні
- articles(code): повертає список статей за даним кодексом, None якщо в рішенні не використовується переданий кодекс