1. Описати абстрактний тип даних, який необхідний для вирішення задачі дослідження

Абстрактний тип даних (АТД) — це тип даних, логічна поведінка якого визначається набором значень та набором операцій що застосовуються цих значень. Вся внутрішня структура такого типу захована від розробника програмного забезпечення — в цьому і полягає суть абстракції. Абстрактний тип даних визначає набір функцій, незалежних від конкретної реалізації типу, для оперування його значеннями. Конкретні реалізації АТД називаються структурами даних.

(<https://uk.wikipedia.org/wiki/%D0%90%D0%B1%D1%81%D1%82%D1%80%D0%B0%D0%BA%D1%82%D0%BD%D0%B8%D0%B9_%D1%82%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85> )

Реалізація проекту буде на Python. В Python наявні деякі абстрактні типи даних.

Можливо створювати абстрактні класи та існують абстрактні типи даних. Створення абстрактних класів допомагають для збереження інформації, для захисту її від пошкодження, для інтерфейсу тощо. Абстрактні типи даних в Python представлені в вигляді списків, словників, сетів тощо. Також можливо використовувати структури даних для виділення пам’яті, тоді реалізація АТД (абстрактних типів данів) буде здійснюватися в вигляді масиву, лінійного списку, стеків, черги, тобто використання динамічного виділення пам’яті.

По-перше, перший АТД який знадобиться для вирішення задачі дослідження – словник. Вся інформація яка буде зчитана з необхідного ресурсу буде записуватись в JSON файл, а там вона буде представлена словнику.

'Словари в Python - неупорядоченные коллекции произвольных объектов с доступом по ключу. Их иногда ещё называют ассоциативными массивами или хеш-таблицами.'

(<https://pythonworld.ru/tipy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovarej.html> )

Робота словника має свої переваги – легко знайти потрібну інформацію по ключу.

По-друге, Отримана інформація буде передаватись в список для подальшого використання її. ‘Списки в Python - упорядоченные изменяемые коллекции объектов произвольных типов (почти как массив, но типы могут отличаться).’

(<https://pythonworld.ru/tipy-dannyx-v-python/spiski-list-funkcii-i-metody-spiskov.html> )

Використання списків має свої переваги в тому, що списки простіше ітерувати. Ця особливість нам знадобиться для фільтрування інформації.

Так як за один запит можливо отримати одну локацію, то потрібно десь її зберігати, для цього можливо знадобиться використання динамічного типу даних, а саме масивів.

'Модуль array определяет массивы в python. Массивы очень похожи на списки, но с ограничением на тип данных и размер каждого элемента.'

(<https://pythonworld.ru/moduli/modul-array-massivy-v-python.html> )

Сам масив базується на списках, так що можливо, для спрощення коду краще використовувати списки.

Використання динамічного типу пам’яті можливо тільки нашкодить роботі програмі, тому що час виконання програми, буде більше, тому що для отримання інформації потрібно буде звертатись до пам’яті та обробка інформації триває доволі довго