**SoundCloud ADT**

SoundCloud ADT – це тип даних, що надає набір методів, які допомагають працювати із даними отриманими при роботі з SoundCloud.

**Структура SoundCloud ADT**

**Ім’я класу:** SoundCloud

При створенні екземпляру даного класу користувач мусить ввести одне слово, яке характеризує його настрій. В подальшому SoundCloud використає дану інформацію для пошуку і підбору пісень.

Після отримання відповіді від користувача, дані будуть записані у змінну self.text. Після цього дана змінна буде використовуватись у методах класу.

**Методи:**

make\_url(self)

make\_playlist(self)

get\_widgets(self)

Опис методів:

*make\_url(self)*

Кінцевий результат: масив із посиланнями на пісні

На початку роботи, якщо змінна self.text не пуста. створюється змінна self.client. Створюючи дану змінну типу client, ми вводимо свій клієнтський id та присвоюємо їй всю необхідну інформацію для роботи із SoundCloud АРІ. Далі за допомогою методу get(‘/tracks’,q = self.text) створюється список з інформацією про пісні (tracks). Наступним кроком є перетворення даних з Resource об’єктів на список кортежів за допомогою методів fields() та items(). Після цього інформація сортується і відбираються лише лінки на треки. Усі посилання записуються до масиву url. Функція повертає вищезгаданий масив url

*make\_playlist(self)*

Кінцевий результат: масив із 4 посиланнями на пісні.

Робота починається із виклику функції *make\_url(self)* і присвоєння змінній url повернутого значення. Далі проводиться перевірка на вміст масиву. Якщо к-сть елементів менша за 4, то наступні дії не виконаються. Наступним кроком є створення масиву playlist та довільного вибору пісень, які будуть наповнювати утворений масив. Функція повертає заповнений масив playlist.

*get\_widgets(self)*

Кінцевий результат: масив із \*віджет-посиланнями.

Спочатку утворюється масив info та викликається функція *make\_playlist(self)* і повернуте значення присвоюється утвореному масиву playlist. Наступний крок – ітерація playlist. Після ітерації отримуємо масив widgets із віджетами. Потім утворюємо масив final і ітеруємо масив widgets. Під час ітерації відділяємо \*віджет-посилання із віджетів і додаємо їх до масиву final. Останній крок повернення функцією масиву final.

**Загальні відомості**

Даний тип даних використовувався при розробці сайту (lofi.pythonanywhere.com) та проекту загалом.

При розробці SoundCloud ADT було використано таку структуру даних як DynamicArray

Також, при створенні SoundCloud ADT було імпортовано такі бібліотеки та модулі, як: DynamicArray (з модулю arrays), soundcloud, random.

\*Віджет-посилання – посилання, які використовуються у HTML віджетах як джерело на відео/аудіо\*