Опис програми для виконання дослідження

1) **collect_artists.py**

Призначення програми:

Накопичити дані про виконавців та зберегти їх у файл для подальшого використання

Вхідні дані програми:

Для коректної роботи програми потрібно вказати spotify client id та spotify client secret, які будуть використані при з'єднанні.

Також потрібно вказати назву директорі всередині artistfiles, де знаходиться інформація про вже проаналізованих виконавців (якщо такої директорії немає, то помилка буде перехоплена і оброблена)

При потребі можна вказати найнижчу популярність артистів, яких буде проаналізовано та максимальну кількість виконавців, яких буде проаналізовано після запуску програми.

Також потрібно вказати назву директорії всередині artistfiles, куди буде збережено інформацію про накопичених артистів.

Структура програми:

Спочатку ініціалізується spotify клієнт, який робитиме запити та повертатиме результати про виконавців.

Згодом ініціалізується об'єкт класу ArtistCollector, який відповідає за збір і збереження інформації про артистів.

Наступний крок – завантаження інформації з директорії, якщо така існує

Потім викликається метод класу get(), який звертається до арі та додає в колекцію інформацію про нових виконавців.

Останній крок – збереження інформації про виконавців до папки в дирекорії artistfiles.

В програмі передбачено перехоплення помилок, таких як переривання вводу або відсутність інтернет-з'єднання. В таких випадках вона припиняє збір інформації, зберігає її до файлу та завершує роботу.

2) collect_albums.py

Призначення програми:

Накопичити дані про альбоми виконавців та зберегти їх для подальшого використання.

Вхідні дані програми:

Для коректної роботи програми потрібно вказати spotify client id та spotify client secret, які будуть використані при з'єднанні.

Також обов'язково потрібно вказати ім'я директорії всередині artsitfiles з уже накопиченими виконавцями (не рекомендовано змінювати файл виконавців

(накопичувати або стирати інформацію), оскільки це призведе до некоректної роботи програми).

Також потрібно вказати назву директорі всередині albumfiles, де знаходиться інформація про вже проаналізовані альбоми (якщо такої директорії немає, то помилка буде перехоплена і оброблена).

При потребі можна вказати найнижчу популярність альбомів, які будуть проаналізовано та максимальну кількість альбомів, які будуть проаналізовано після запуску програми.

Також потрібно вказати назву директорії всередині albumfiles, куди буде збережено інформацію про накопичені альбоми.

Структура програми:

Спочатку ініціалізується spotify клієнт, який робитиме запити та повертатиме результати про виконавців.

Згодом ініціалізується об'єкт класу AlbumCollector, який відповідає за збір і збереження інформації про альбоми.

Наступний крок – завантаження інформації з директорії артистів та альбомів, якщо така існує.

Потім викликається метод класу get(), який звертається до арі та додає в колекцію інформацію про нові альбоми.

Останній крок – збереження інформації про альбоми до папки в дирекорії albumfiles.

В програмі передбачено перехоплення помилок, таких як переривання вводу або відсутність інтернет-з'єднання. В таких випадках вона припиняє збір інформації, зберігає її до файлу та завершує роботу.

3) **collect_songs.py**

діє аналогічно до collect_albums.py

Відмінність:

- завантажується інформація про проаналізовані альбоми та проаналізовані пісні(не рекомендовано змінювати файл альбомів (накопичувати або стирати інформацію), оскільки це призведе до некоректної роботи програми).
- Інформація про проаналізовані пісні зберігається в папці, яка знаходиться в songfiles

4) **year_coverage.py**

Призначення програми:

Визначити, які роки покрила вибірка пісень, яка буле сформована

Структура програми:

Спочатку ініціалізується контейнер з альбомами. Для цього відкривається файл з текстовим представленням альбомів.

Згодом ітеруючи по альбому, в множину записуються рік випуску кожного альбому.

Ці значення виводяться на екран.

5) **Form_report.py**

Призначення програми:

Сформувати доповідь щодо усіх проаналізованих років та записати їх у файл

Вхідні дані програми:

Для коректної роботи програми необхідно попередньо проаналізувати виконавців, альбоми й пісні та відкрити ці файли для читання.

Структура програми:

Спочатку з відкритих файлів про артистів, альбоми та пісні формуються контейнери.

Потім створюється екземпляр класу TimeRange, який відповідає за певний відрізок часу.

Наступний крок – почергове додавання усіх пісень до TimeDelta, який сам визначить, коли добавляти дану пісню в залежності від року релізу

Останній крок – запис звіту за даний відрізок часу у файл