4.3

Описати програму, яка розроблена для виконання завдання дослідження. Опис повинен містити такі основні пункти:

- Призначення та коротка характеристика програми

- Вхідні та вихідні дані програми

- Структура програми з коротким описом модулів, функцій, класів та методів

. - Коротка інструкція по користуванню програмою

- Опис тестових прикладів для первірки працездатності програми

Нагадую ідею: у мене є база даних книг, які містять певні характеристики. Програма запитує користувача, які приблизно книги його цікавлять і підбирає найбільш відповідні.

Програма використовує не лише iTunes Search API, тому що він дозволяє отримувати інформацію про продукт чи автора лише тоді, коли користувач вводить конкретну назву. Тож накопичити базу даних майже неможливо (майже, тому що по одному вводити назву книги нераціонально). Отож, я використала ще одне API: Goodreads API. Ознайомившись з даними, які мені може надати це API та як його використовувати, я зайшла на сайт <https://www.goodreads.com/api> та отримала ключ до нього. У модулі “list\_books.py” знаходиться реалізація отримання списку книг, які я за допомогою вікна команд переміщаю отримані дані у файл “book.txt”. (python list\_books.py > book.txt).

Переходимо до модуля під назвою «analyse\_books.py». У цьому модулі я створила клас Books. В конструкторі \_\_init\_\_ мені немає потреби ініціалізовувати змінних, тому я написала просто pass.

Наступний метод – read\_file(). Він зчитує з файлу “books.txt” список книг та записує його у список lst\_of\_books. Я зчитую саме з файлу “books.txt”, а не “book.txt”, тому що у “book.txt” програма записувала мені потрохи книги і, оскільки програма щоразу перезаписує у файл заново, то я копіювала частину книг і переміщала їх у “books.txt”, тобто більший файл.

Наступний крок – метод find\_book(). У цьому методі програма виконує цикл, якому проходиться по кожній назві книги, яка міститься у списку (отриманий у попередньому методі) і надсилає запити за допомогою iTunes Search API, щоб визначити, існує така книга у базі даних iTunes Store, чи ні. Перевіряю за допомогою атрибуту type. Як вже згадувалось раніше, кожен продукт має про себе всю інформацію. Кожна книга має у собі «характеристику» type, який називається “ebook”. Якщо такий тип є в інформації про книгу, значить це книга і вона там взагалі є. Потім програма додає назву книги, автора, ціну, жанри і посилання на цю книгу в iTunes в окремий список, який називається personal\_lst. Це окремий список для кожної книги і інформації про неї. Потім той список програма додає в загальний список (списки в списку). Personal\_lst програма спустошує, щоб інформація про наступну книгу туди надійшла. Ось так працює цикл. Оскільки далеко не кожна назва в списку lst\_of\_books дає позитивний результат після запиту, то збуджується виняток LookupError. Тому я використала конструкцію try/except. Цей метод повертає список списків з книжок з потрібною інформацією про них.

Далі метод genres(). Він окремо обробляє жанри книг, які є останнім елементом в списку, який повертає попередній метод. Окремо, тому що жанри в книгах повторюються і вони знаходяться в окремому списку. Тому я створила множину, куди програма додає усі жанри, уникаючи повторень. Зрештою, програма виводить список жанрів і пропонує користувачу вибрати один з них. Повертає список жанрів.

Останній і головний метод – analyse\_data(). Програма запитує користувача ввести ціну книги, бажану для нього. Якщо користувач хоче безкоштовну – потрібно просто ввести «0». Після цього ще одне питання: Який жанр користувач хоче? Програма аналізує отримані дані і повертає перелік книг, які б могли підійти користувачу. Разом з назвою і автором є також ціна, жанри і посилання на книжку на сайті, якщо вона зацікавила користувача.

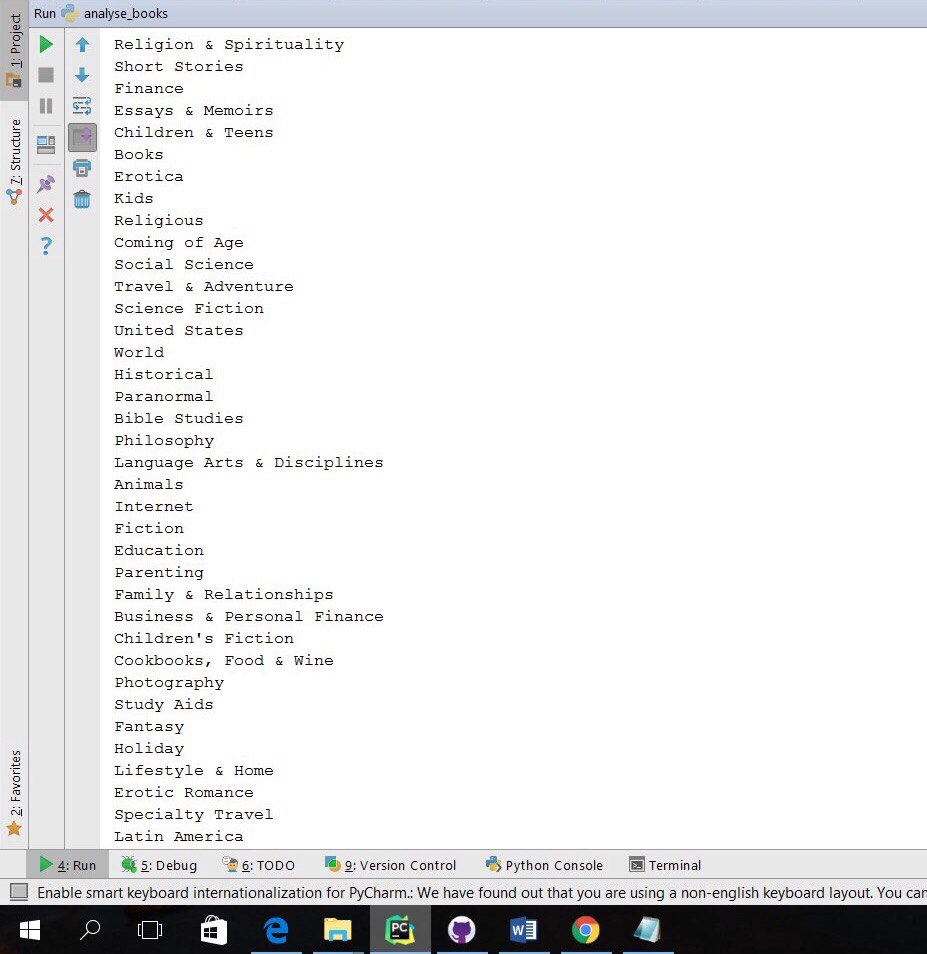
Вхідні дані з боку користувача: число, яке означає ціну; слово, що означає жанр.

Вхідні дані з боку програми: база книг у файлі.

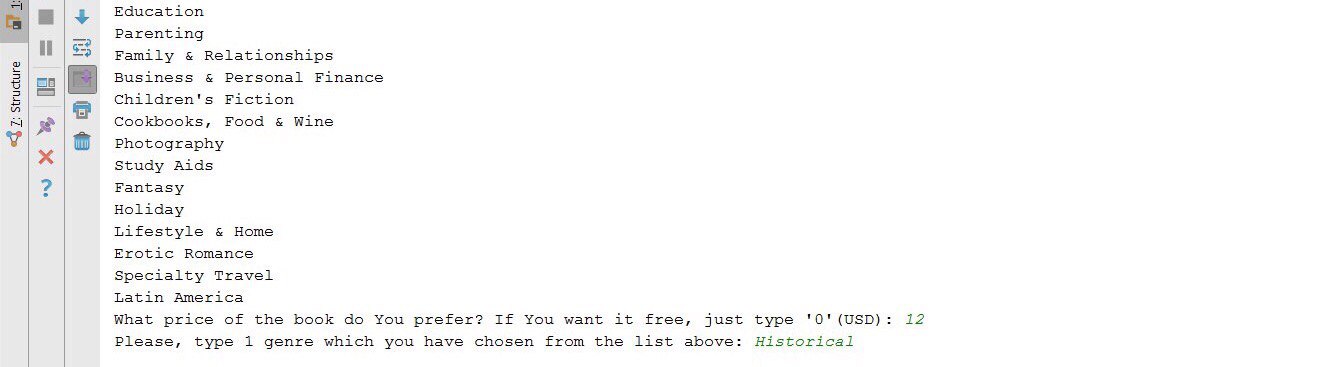
Вихідні дані: перелік книг з потрібною інформацією.

Приклад:

Перелік жанрів:



Приклад відповідей на запитання:



Вивід:

