Case Books, Games, Videos, and other assorted Media

You are given the task to implement a system for keeping track of books and entertainment media. To begin with, you will support keep track of books (as you've done in previous exercises).

You will develop a system that keeps track of books, videos (movies, series, etc.). Your system should support all 3 categories described above.

A user should be able to:

- Start your program and get some interface prompt and a welcome message, with some options that the user can perform (using System.out and Scanner).
- Select a type of media that they want to see. The media supported should include at least books, videos, games. You can add more media that you are interested in.
- Each media has some connection to some people (authors of books, directors and actors in videos, producers and/or storywriters for games).
- See what items there are for that type of media in your collection.
- Select and modify one of the items in the collection.
- Find an item in the collection based on title or other characteristics.
- Add new items of a particular type.
- Remove an item.
- Exit the program. (When the program ends, all changes will be written to a file).

In addition, a users should be able to:

- Search for an item by title, or people involved. The search should return all media that has a match for the title (this includes a partial match), or a match for the names of people involved.
    - Example: searching for "dune" should return
      "Dune" – the book by Frank Herbert
      "Sandworms of Dune" – the book by Brian Herbert
      "Dune" – the film directed by Dennis Villeneuve
      "Dune: Imperium" – the boardgame designed by Paul Dennen

All the relevant data should be stored in a database. Additions and changes should be made in such a way that user inserted data that is valid should not be lost, even if the program ends unexpectedly.

The program should not be vulnerable to SQL injection.

Delivery:

Deadline: February 9

Contents:

- An archive that contains the source files needed to run your code.
- NOTE: Your archive should not contain
- A text file (.doc, .pdf) where you describe (briefly 200-400 words) how you approached the problem.
    - What concepts from OOP were useful to solving this problem, how did you approach your solution?
    - How did you structure your database and domain model?
    - How did you handle searching for several types of media in one search and why did you approach the problem in that manner.

Advanced Step:

If you are done with the task above (and only if you are done with the task), consider more advanced searches. For example, searching for items that match a term, but restricted to only some of the categories.

Example: searching for “dune”, and selecting only “books” and “games” should return

- “Dune” – the book by Frank Herbert
- “Sandworms of Dune” – the book by Brian Herbert
- “Dune: Imperium” – the boardgame designed by Paul Dennen

While searching for “dune”, and selecting only “books” and “video” should return

- “Dune” – the book by Frank Herbert
- “Sandworms of Dune” – the book by Brian Herbert
- “Dune” – the film directed by Dennis Villeneuve

When designing this upgraded search, think about how it would change if/when a new category is added.

Some tips:

- Do not try to make everything at once. When creating a program, it will be an advantage to start with something simple and build on, little by little. Check that what you make works along the way. A sensible first step could be, for example, to create a program that manages to display a menu. A next step may be to try to read the books when the program starts.

- This is not a work requirement, so you can easily opt out of some functionality. Make as much as you can with the time you have. You can also choose to implement something other than what the task asks for. The most important thing is that you spend time writing code. **Remember that this does not apply for the exam. Requirements there are more inflexible.**

- Some may have made chapters in the books in an extra assignment before. Bring it if you want. Remember that you must then enter information about chapters in bok.txt.

- Remember to try out your code before submission. Does it still work? Have you moved files around (or renamed them) that may cause the code to not run?

- remember to include either code or sql scripts that setup the database structure that you want to use. It may be hard to replicate your database structure (and thus get your program running), if that information is missing.