PROJECT ASSIGNMENT (Java 1)

Create an application that allows updating data about connected entities, as desired. In the following, an example of the **Movie** entity and associated entities **Actor**, **Director** and the like will be selected for the entities, which you can freely take as your own.

When building the application, it is important to respect the best principles of the object-oriented paradigm and using libraries (**Class Library**), according to the following instructions:

- The application saves data in the **Microsoft SQLServer** database
- It is necessary to create an initialization **DDL** script to create all the tables that will be used in the application
- It is necessary to create a script to delete all data from the tables
- All work with the database takes place by calling procedures from Java code, using the **Repository** pattern
- The application contains several types of users, so it is necessary to provide 2 roles - **Administrator** and **User**
- The **Administrator** must be at least one user with the associated username/password pair that will be created using the procedure after the initialization script
- Access to the application is restricted to the **Login** form, where simple user registration of the **User role** is also possible
- The application is put in the correct state for use by the **User** by the **Administrator**
    - When entering the application, the **Administrator** has the option of deleting all data from the database (which also deletes all images from the data system) and uploading new data to the database, by calling the **RSS parser** - this represents a simple and complete administrator interface

- **RSS parser** is an application component that parses all XML data from a specific URL and saves it in the database for later use
  - Example: **https://www.blitz-cinestar-bh.ba/rss.aspx?id=2682**
  - Images must be downloaded to a local directory (example: **assets**) and their relative paths must be saved in the database
  - Examples of RSS feeds:
    https://blog.elink.io/top-world-news-websites-and-rss-feeds/
    https://www.uen.org/feeds/lists.shtml
    https://www.nasa.gov/content/nasa-rss-feeds

- After entering the application, the **User** is presented with a form for viewing and changing entities (**CRUD** operations), in this example **Movie**

- Forms within the application must be well organized (example: **JTabbedPane**, where each **JPanel** is represented by its own class according to the **Single-responsibility principle**)

- It is necessary to create additional entities (example: **Actor**, **Director**) that also have **CRUD** forms

- The existing entity needs to be updated in such a way that it can be connected to new entities (example: **Movie** can be connected to several **Actor**, **Director** entities)

- When changing an entity or deleting it, it is necessary to ensure a consistent change / deletion of the associated images from the image directory (example: **assets**)

- To display entities, it is necessary to use the **JTable** and associated **AbstractTableModel** models to the greatest extent

- To navigate in the application, it is necessary to use **JMenu**

- It is necessary to implement **Drag and drop** functionality on your own selected example in the application (example: adding **Actor** to **Movie**)

- It is necessary to implement the **XML download** of any entity using the **JAXB library**