OOP Project 3: XML Movies Manager (Individual Project)

Due Dates:
Part 1: 2020-10-09
Part 2A (50% of the functionalities finished): 2020-10-18
Part 2B (100% of the functionalities finished): 2020-10-23
Part 3: 2020-10-25

Overview:
In this individual project, you will continue to reinforce the basic OOP concepts learned in the OOP 1 course and you will apply the new concepts learned in the OOP2 course. You will also build a program following a simple development process. You will improve the quality of the program by performing a series of tests and bugs fixings. Finally, you will perform some research on Internet to learn how to add new features to your program.

The project:
You are required to build a program named *XML Movies Manager (XMM)*. When the program starts, it must read an XML file containing movies information. When the program closes, the movies information must be saved in the same XML file. When the user selects a particular movie, all its information must be displayed and also its corresponding image (the image path is indicated in the XML file but you must change it depending on the actual path in your project).

Here are the required functionalities of this program:
- Display the different movie genres found in the XML file (genre repetitions are not allowed).
- For the selected genre, display the corresponding list of movies. That list must contain the name and the year of each movie (you must use a 2-columns ListBox).
- For the selected movie: display its info plus its image.
- For the selected movie: modify its info and its image.
- For the selected movie: delete de movie (this won't delete its image).
- Add a new movie:
  - The user can select the movie genre from the genre list. The user could also define a new genre when adding a new movie.
  - The user can place the movie image in the image folder before adding the image in the program. To make things easier, use only images having the following dimensions: Width: 300 pixels, Height: 400 pixels
  - *Bonus:* the user selects the image in the file system and the program copies the image to the image folder.
- *Optional:* Search a movie by its name or by its year (display the movie info if found, otherwise, display an appropriate message)

When the program exits, the movies list must be saved to the XML file

Development Process:

Part1: GUIs, Class Diagram and Git
In this part, you will produce the *GUIs* and the *Class Diagram* for this program. The GUIs must correspond to the required functionalities of the project.
The VS Project must be linked to your GitHub account. At least the Master Branch must be defined.

The class diagram must include:
- The name and type of each class.

- The properties of each class.
- The methods and constructors of each class.
- The links (relationships between classes; e.g. inheritance relationships)

Part2: Coding

In this part, you will code the program. Your code must meet the following conditions:

- Use the OOP approach learned in the OOP 1 and OOP 2 courses.
- **Use a single list of movies.**
- Follow the Coding Standards and Naming Conventions explained in https://www.dofactory.com/reference/csharp-coding-standards
- Add appropriate XML comments to each class and method. Add comments to complex blocks of code.
- Use GIT to manage the files versions of your program.
- Avoid the code *copy/paste*, this is a bad practice. This can be avoided by using appropriate *Class Methods*.

Part3: Program testing and Class Diagram update

In this part, you must test your program and solve all the bugs found. You must also produce the final version of the class diagram. At the end of this part, you program should be unbreakable.

Project Submission:

- A title page is required for each submission.
- The GUIs and the Class Diagram must be submitted in a single PDF file. Each element must be identified by a name.
- The code source of the program must be submitted in a single .zip file. That file must contain the VS Solution of the project and must run as is (test the code submission before submitting it to the teacher).