OOP Project 2: Personal Music Collection (Team Project)

Due Dates:
Part 1: 2020-09-27
Part 2A (50% of the functionalities finished): 2020-10-01
Part 2B (100% of the functionalities finished): 2020-10-05
Part 3: 2020-10-07

Overview:
In this team project, you will refresh and reinforce more OOP concepts learned in the OOP 1 course. You will also get used to build a program following a simple development process. You will use Git as a Version Management Tool. Finally, you will improve the quality of the program by performing a series of tests and fixing bugs.

The project:
You are required to build a program to list the files located in a Directory. The user must specify (or select) the directory to analyse. The program must display the list of files found in the directory. For each file, the program must display the:
- File Name
- File Type (extension)
- File Size
- File Last Modification Date/Time

The program must offer the possibility to reorder the displayed files by:
- File Name
- File Type
- File Size
- File Last Modification Date/Time

The program must also display:
- The number of files found in the directory
- The number of files per type

The types of files your program must support are:
- *Document files:* pdf, docx, txt
- *Image files:* png, jpg, bmp
- *Media files:*
  - *Video files:* mp4, avi, mkv
  - *Audio files:* mp3, aac

***Other types of files must be considered as “Other Files”

When the user selects a file from the list, the program must display the properties of the selected file.
Here are the file properties to display:

Any File	Document Files	Image Files	Media Files	Video Files	Audio
- File Name and Extension - File Size - Creation Date/Time - Modification Date/Time - Comments on file	- Number of pages - Number of words - Doc Subject	- Width - Height - H. Resolution - V. Resolution	- Title - Length - Rating	- Director - Producer	- Artist - BitRate

The properties in brown are user-defined properties.
The properties in black are generated by the OS.
Bonus: for audio files, their properties can be found in their ID3 tags (then the user can define the *Artist* property)

Your program must allow the user to set, modify and delete the user-defined properties of each file.

Finally, an "Export List" button will generate a text file containing the information about each file in the directory. Th program must as the user the desired path/name of the text file.

Development Process:

Part1: GUIs and 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 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)

The VS Project linked with the Git environment of the team must be created. At least the Master Branch must be defined and both teammates can collaborate in the project.

Your teacher must approve the Part 1 before starting the next part.

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 course. **You must use the Inheritance behaviour of OOP.**
- Follow the Coding Standards and Naming Conventions explained in https://www.dofactory.com/reference/csharp-coding-standards
- Add appropriate comments to each class, method and to complex blocks of code. You must follow the XML format learned in class. You must use DocFX to generate the project documentation in Web format.
- 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).