TP#3: January 18th – Collection, LINQ, threads

The deadline for this TP is tomorrow evening, midnight. It is a small TP that we'll use to make the link between collections, which are used to store and expose data in C#, and *LINQ*, a syntax language that will be helpful in formatting this data properly.

Exercise 1:

Get the *MovieCollection.cs* file in the TP3_Ressources folder. It contains a collection of almost all Disney movies ever made. This data collection is a typical example of what you can find online: This collection is incomplete, contains missing or wrong data… but you'll have to work with it, no matter what. Unless stated otherwise, creating new class or variable is forbidden. Using if/else statement is not the way to do this. All these requirements should be single statement, two at most.

Using *LINQ*, you'll have to (and print):

- Display the title of the oldest movie.
- Count all movies.
- Count all movies with the letter **e**. at least once in the title.
- Count how many time the letter **f** is in all the titles from this list.
- Display the title of the film with the higher budget.
- Display the title of the movie with the lowest box office.
- Order the movies by reversed alphabetical order and print the first 11 of the list.
- Count all the movies made before 1980.
- Display the average running time of movies having a vowel as the first letter.
- Print all movies with the letter **H** or **W** in the title, but not the letter **I** or **T**.
- Calculate the mean of all Budget / Box Office of every movie ever

Optional, harder questions:

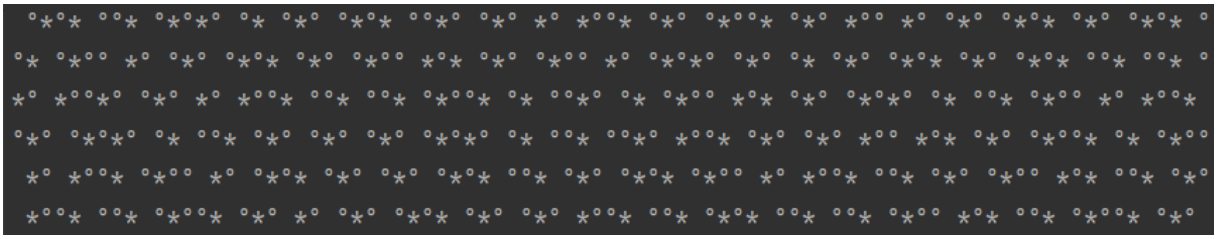
- Group all films by the number of characters in the title screen and print the count of movies by letter in the film. i.e.:
  - 3 char => 5 films
  - 4 char => 8 films
  - 5 char => 1 film
  - (…)
- Calculate the mean of all Budget / Box Office of every movie grouped by yearly release date
  - 1941 => $123456
  - 1942 => $123123
  - (…)

Exercise 2:

Create a simple function that create 3 threads:

1. The 1st prints an empty space for 10 seconds, every 50ms.
2. The 2nd prints * for 11 seconds, every 40ms.
3. The 3rd prints ° for 9 seconds, every 20ms.

There should only be ONE FUNCTION that can print, and it should be accessed by a mutex (Google it). The expected result should look like a night sky, full of stars.



Free time:

Once both exercises are done, feel free to leave, show me your projects, or ask me any questions you want that are work/computer/trading related. I'll gladly help you during this timeframe. This will also help me evaluate your trading and financial knowledge to anticipate for the more theory-oriented classes that will arrive at the end of the semester.

Bonus information:

The exam will be written by me, and the class will be divided in groups. The topics are:

- A bit of algorithmic
- Loops and conditional structures
- Collections
- Asynchronous programming
- Some trading stuff that I would have modelized for this exam and that I'll ask you to work on, but we'll get to this part later in this course.

It is going to be only a machine exam, no paper, no need-to-know stuff by heart, in a ~2h timeframe. Questions? Please ask!