VGP 232 Game Tools and Pipeline

# Assignment 2a

## Console Application (improvement)

## Objective:

## Refactoring

Using your existing assignment 1 **C# Console Application** improve it by refactoring:
- Extending the Pokemon class (i.e. Pokemon.cs)
    - Implement this method to Pokemon
      static bool TryParse(string rawData, out Pokemon pokemon)
      should similar to float.TryParse(string, out float)
    - Also add an **enum** member **PokemonType** containing the following: Grass, Fire, Water, Bug, Normal, Poison, Electric, Ground, Fairy, Fighting, Psychic, Rock, Ghost, Ice, Dragon
    - Override the ToString() method so it can print the pokemon back in the CSV format
- Add a PokeDex class (i.e. PokeDex.cs) which inherits from List<Pokemon>
  i.e. public class PokeDex : List<Pokemon>
  implements the following interfaces:
    - Pokemon GetHighestDefense()
    - Pokemon GetHighestAttack()
    - Pokemon GetHighestHp()
    - Pokemon GetHighestMaxCp()
    - void PrintSortBy(string name)
    - void Save(string filename)
    - PokeDex Load(string filename)

Refactoring:
- Move the parse functionality from the Program.cs main method to the Load function of the PokeDex class
- The Pokemon class will have a TryParse(...) mentioned above which Load will call and handle any exceptions that may occur when the values that are being parsed is invalid i.e. invalid number of rows, wrong types to convert etc.

Additional Feature:
- Add an option to add a new Pokemon to the list (interactive mode)
- Add an option to edit an existing Pokemon's properties (HP, Defense etc) (interactive mode)

VGP 232 Game Tools and Pipeline

- Add an option to remove existing pokemon (interactive mode)
- Add an option to save the edited list into csv format (interactive mode)
- Add an option to sort the Pokemons by name alphabetically i.e --sort-alpha or -sa
- Add an option to sort the Pokemons by the hp, attack, defense or, max cp
    - using --sort <type to sort by> or -s <type to sort by>

## Unit Test:

Add nUnit tests to confirm that your class can do the following:
- PokeDex
    - Find highest defense
    - Find highest hp
    - Find highest attack
    - Find highest max cp
    - Load a pokemon.csv file
    - Save the csv file output
        - Note for this, you may need to use the SetUp and TearDown step so it runs it cleanly
- Pokemon
    - TryParse valid values
    - TryParse invalid values

Error Handling:
- invalid path (path does not exist)
- invalid arguments
- invalid data from the csv

C# Helpful links

https://support.microsoft.com/en-us/help/816149/how-to-read-from-and-write-to-a-text-file-by-using-visual-c

https://www.dotnetperls.com/sort-list

# Due date

Next class, week 3 (at the start of class 6:30pm)

VGP 232 Game Tools and Pipeline

## Submission

Implement the answers in a new project with the name "Assignment2". This project should be submitted through committing and pushing through GIT to your VGP232 repository which you shared with the instructor.

## Grading

Marks: Out of 100 (10% of final grade)

(60) Functional: Does it compile? Does it meet the requirements and work? Does it give the correct results?

(25) Error Handling: Does you application handle bad input? If so, does it handle failures and exceptions gracefully or does it crash?

(15) Naming convention & Comments: Does it follow the coding standards? Are your variables and method names descriptive? Did you leave descriptive comments on methods that does not have obvious functionality?

## Coding Standard

**C#**
https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions

## Late Assignments

After the due date, the student will no longer be able to submit their assignment and will receive a 0 as I will go over the solution in the next class.