TRENT UNIVERSITY DURHAM GREATER TORONTO AREA

COIS 2020 H: Data Structures and Algorithms

2024, Winter

Assignment 1 (12%)

Overview:

In this assignment, you will create a series of objects, that will have a position (which is itself an object). You're going to store collections of these objects in an array and a list, and you will need to perform basic operations on those structures.

Required Classes:

1. Class **`Position`**, which will have
   - doubles `x, y`, and `z` with setters and getters. These values will be limited to the range +/- 15.0 (i.e., from -15.0 to 15.0). If the user enters a value outside that range, clamp to the closest value.
   - Method `move(),` which takes doubles `dx, dy`, and `dz` and changes the position by +`dx`, +`dy`, +`dz` (it clamps to max values of +/- 15 for the edges). The three values `dx, dy`, and `dz` can be positive or negative.

   .

2. Class **`Animal`**, which will have
   - Properties: `int ID, string name, double age, Position pos`.
   - Methods.
     - `Move` method. It passes inputted (`dx, dy, dz`) to the position class (which also will have a move method, the move method in ***`Position`*** should handle the actual implementation of moving things).
     - Setters and getters for the fields/properties,
   - `ToString`, (`public override string ToString()`). The Animal class **and** *each subclass* will implement its own `ToString()`, and it should just return all of the properties (including, `ID, name, age, pos`) in some sane format.

3. Animal subclass: **`Cat`**
   - `Enum` property `Breed` {Abyssinian, British Shorthair, Bengal, Himalayan, Ocicat, Serval} (if this confused you, look up Enumeration Types on the MSDN[1] C# reference).

4. Animal Subclass: **`Snake`**
   - Properties: `double length, bool venomous`.

[1] Enumeration types - C# reference | Microsoft Learn

Main method:

1. Generate a series of animals, 3 cats and 3 snakes, with random properties. The names must be selected automatically (programaticaly) and randomly from the text files provided, and you should avoid duplicates. [Hint: check the Appendix for example about reading data from a file]
2. Load the created animals (all of them) into an array **and** a list (using the .Net `ArrayList`). [Hint: to use a built-in List, you will need to add: "using System.Collections.Generic;"]
3. Traverse each collection (the array and the list) and print off the properties of each object. A `foreach` loop is fine. This must be a loop with a single line.
4. Move all objects with a **random** dx, dy, dz (but only slightly; these can be +/- 2.0 but restrict the final position to +/- 15). [Hint: the following code generates and prints five random numbers between 0 and 10].

```
var rand = new Random();
for(int i = 0; i<5; i++)
     Console.WriteLine(rand.Next(0, 10));
```

- Print off all objects and positions again and highlight in your screenshot that the move works.

IMPORTANT:

- As this is not an introduction course, documenting any code is trivial to you. So, all your codes **must** be documented sufficiently. Your source code MUST have comments explaining what each class and each method is for, etc.

Submission guidelines:

- Fill out the Submission Template file provided with the assignment.
- Name it for your ***trentusername.docx***.
- Save it as a .pdf file.
- Upload the following to the assignment link on Blackboard:
  - The .docx file
  - The .pdf file
  - a zip file (named for your trentusername.zip) containing your Visual Studio project directories.

APPENDIX: READING FROM FILE

Following is one way to read the information from a text file in C#.

```
      string[] text = System.IO.File.ReadAllLines(path);//Path should be replaced
with your file’s path

      string[] Names = new string[text.Length];
           for (int i = 0; i < text.Length; i++)
           {
               string[] partial = text2[i].Split('.');
               Names[i] = partial[1];
            }
```