CS 2026 – Spring 2010
Assignment 3
2/13/2010
Due: 2/21/2010 11:59 PM

This assignment will exercise your knowledge of C# features such as reflection, exceptions, LINQ and threading that were discussed in class.

In this assignment you will implement a Querier class that will take in a collection of objects and answer basic queries about them. A Querier object is initialized with a collection of items that it stores internally and then provides an interface to query them.

You will not know the type of items given to your Querier in advance. All you know is that each item will have one or more properties, and each property will be of type either: int (System.Int32), bool (System.Boolean), or string (System.String).

You will use reflection to manipulate the elements of your collection. Additionally, elements might have some methods that will change the value of some of their properties.

All the queries should be written in LINQ. Queries not written in LINQ (i.e. written with traditional loops) will not be accepted and will fail the assignment.

Methods

In all the following methods, properties names and values are passed in as strings. The same goes for functions to apply.

When dealing with passed-in property and function names you will need to use reflection to access those properties and functions. If the method also takes in a value for a property, you will have to use reflection to determine the type of the property and then you should convert the passed-in property value to the appropriate type and use that in your LINQ expression.

For example, if the type of the property turns out to be an integer, you should use its value in the LINQ query as follows: int value = int.Parse(propertyValue);

If a given property name does not correspond to a property of the elements in the collection then you will get an exception and you should catch that exception and return -1. Similarly, if the passed-in function name does not match a method of the elements of the collection, or the passed-in property value does not match the property type then you should catch the resulting exception and return -1.

Your Querier class should implement the following methods:

- Querier(IEnumerable<object> e)
    - A constructor that takes in the collection of objects that will be operated on.

- public int CountByProperty(string propertyName, string propertyValue)
    - This method returns the number of elements in the collection that have the specified value for the specified property.
- public int SumByProperty(string propertyName1, string propertyValue1, string propertyName1, string propertyValue1)
    - The return value is equivalent to CountByProperty(string propertyName1, string propertyValue1) + CountByProperty(string propertyName2, string propertyValue2). You must use two threads to compute the sub result. Your main thread will wait for the two child threads until they both get the result. Then you can return the sum of the two sub results.
- public int GroupByProperty(string propertyName)
    - This method will take in the name of a property and print out to the console how many elements in the collection have the same value for this property.

      So if the passed-in property name is a car's manufacturing year, the console printout should show how many cars were made in each year. All you have to printout is a table with two columns that looks like this

      property value - - # of elements that have that value

      The printout should be ordered in descending order by the property value.

      An example output will be something like this:
      2009 6
      2008 12
      2007 10

      This method should return 1 if the given propertyName is for a valid property, -1 otherwise.
- public int Transform(string functionName)
    - This method applies the functionName for all the collections elements. As always, this function returns 1 if functionName is a valid method of the collection's elements, -1 otherwise.
    - You do not need to use a LINQ expression in this method.

Save your program in a single file called Program.cs and submit it to CMS.

Good luck!