Assignment 2

Background:

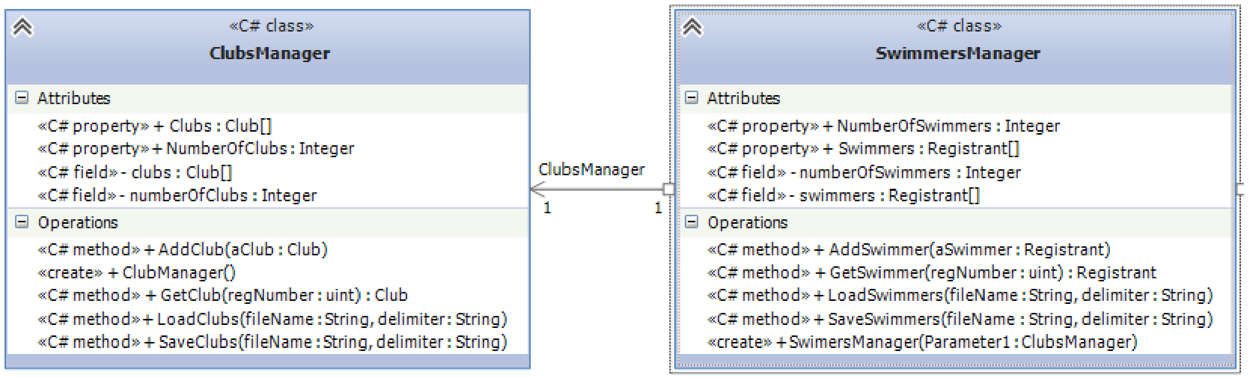
You are creating application that is going to track the swim clubs, their swimmers, swim meets and results.

References: Please refer to “General Assignment Requirements” document posted on eCentennial.

Part 2:

Modify Assignment 2 to satisfy the following:

Add classes with their members that are represented by following class diagram:


«C# class»
ClubsManager
Attributes
«C# property» + Clubs : Club[]
«C# property» + NumberOfClubs : Integer
«C# field» - clubs : Club[]
«C# field» - numberOfClubs : Integer
Operations
«C# method» + AddClub(aClub : Club)
«create» + ClubManager()
«C# method» + GetClub(regNumber : uint) : Club
«C# method» + LoadClubs(fileName : String, delimiter : String)
«C# method» + SaveClubs(fileName : String, delimiter : String)
ClubsManager
1
1
«C# class»
SwimmersManager
Attributes
«C# property» + NumberOfSwimmers : Integer
«C# property» + Swimmers : Registrant[]
«C# field» - numberOfSwimmers : Integer
«C# field» - swimmers : Registrant[]
Operations
«C# method» + AddSwimmer(aSwimmer : Registrant)
«C# method» + GetSwimmer(regNumber : uint) : Registrant
«C# method» + LoadSwimmers(fileName : String, delimiter : String)
«C# method» + SaveSwimmers(fileName : String, delimiter : String)
«create» + SwimersManager(Parameter1 : ClubsManager)

Explanation of the classes and members:

ClubsManager class:

1. *clubs* is a field that contains a list of *Clubs*. Assume there will be no more than 100 clubs. *Clubs* is a corresponding property.

2. *numberOfClubs* is a field that contains a number of clubs in the Clubs list. *NumberOfClubs* is a corresponding property.

3. *GetClub* retunes a *Club* that has registration number that is specified by the parameter. If a club is not found, it returns null.

4. *LoadClubs* method loads and add all *Clubs* to the *clubs* field from a file specified by *fileName* parameter. The fields in a record are delimited by the string specified by *delimiter* parameter.

5. Clubs text file is comma delimited file with the following fields in order:

   1. Registration number – mandatory field
   2. Club name – mandatory field
   3. Street address – optional field
   4. City address – optional field
   5. Province address – optional field
   6. Postal code address – optional field
   7. Phone number – mandatory field

6. *SaveClubs* method saves all the clubs in the *clubs* field to the file with structure described in point 5. The file name is specified by the *fileName* parameter and the fields are delimited by *delimiter* parameter.

7. AddClub method adds a club to the field *clubs.*

SwimmersManager class:

1. *swimmers* is a field that contains a list of *Registrant*s. Assume there will be no more than 100 swimmers. *Swimmers* is a corresponding property.

2. *numberOfSwimmers* is a field that contains a number of swimmers in the swimmers list. *NumberOfSwimmers* is a corresponding property.

3. *GetSwimmer* retunes a *Registrant* that has registration number that is specified by the parameter. If a swimmer is not found, it returns null.

4. *LoadSwimmers* method adds *Registrants* to the swimmer field that are loaded from a file specified by *fileName* parameter. The fields in the record are delimited by the string specified by *delimiter* parameter.

5. Swimmers text file is comma delimited file with the following fields in order:
   1. Registration number – mandatory field
   2. Swimmer name – mandatory field
   3. Date of birth – mandatory field
   4. Street address – optional field
   5. City address – optional field
   6. Province address – optional field
   7. Postal code address – optional field
   8. Phone number – mandatory field
   9. Club registration number – optional field – club that swimmer is registered with

6. *SaveSwimmers* method saves all the swimmers in the *swimmers* field to the file with structure described in the previous point. The file name is specified by the *fileName* parameter and the fields are delimited by *delimiter* parameter.

7. *AddSwimmer* method adds a swimmer to the field *swimmers.* If the swimmer has been assigned a club and the club is not in *clubs* field of *ClubsManager* it is added to the *ClubsManager*.

Additional requirements:

1. Make sure that all classes except Program class are in a class library.
2. The test harness is provided in attached program.txt file. You must not modify the content of the class. You can change namespace names only.
3. Modify your code from assignment 2A to provide the similar output as in the text file output.rtf
4. The two text files (Clubs.txt and Swimmers.txt) that are provided contain the clubs and swimmers that need to be loaded.
5. The two files that are generated by the application are also provided for the reference.
6. You can add any additional code that may be needed.

Submission: Submit the solution to Assignment2 drop box by drop box deadline.