Application Development with .NET (32998, 31927)

Lab -9 Questions

Please download the sample code from Canvas and follow the instructions

Program 1:

Write a program to read the employee details from a text file and store them in a List. Use the following class diagrams as a reference:

Employee: : IComparable<Employee>
+employeeFName: String +employeeSName: String +hourlyRate: double +workHours: double +employeeID: int
-Employee () -LoadEmployee(string fileLine):void -GetWeeklySal(): double -ToString(): string -CompareTo(Employee other):int

EmployeeList
+ List<Employee> : employees
- EmployeeList () - LoadEmployees(string filename): void - PrintEmployees():void - SortEmployees(): void

Method Description:

1. **Class: Employee (Separate .cs file but same namespace)**
   a. **LoadEmployee**(): Method receives emp.txt filename as a parameter and uses **StreamReader** to read the file content. Passes a line single data line to the LoadEmployees() method of EmployeeList class for further processing.
   b. **GetWeeklySal**(): Calculates and returns the weekly salary of an employee based on the number of working hours per week and the hourly pay rate.
   c. **ToString**(): Overide the ToString() method to which returns the employee details in the following format:

< employeeFName> <employeeSName> ID#:<employeeID> Weekly Income: <GetWeeklySal()>

   d. **CompareTo**(): Compares employeeID values and
   - returns -1: if the first ID is less than the other.
   - returns: 1: if equal
   - else returns 0

2. **Class: EmployeeList (Separate .cs file but same namespace)**
   a. **EmployeeList**(): The constructor creates the instance of List<Employee>
   b. **LoadEmployees**(): Method receives a single line of comma-separated data from them as a string and extracts individual fields from the line by splitting the line using ',' delimiter. The extracted fields are then assigned to respective data variables.

c. **PrintEmployees**(): Display the details of all employees by calling the ToString() method on each Employee object present in the list.
d. **SortEmployee**(): Sorts the employee details based on the Employee ID. (Uses the IComparable interface.)

Text file: emp.txt format:

<FirstName>, <LastName>, <hourly rate>, <employee ID>, <working hours per week>

Example:

Jetson, Jordon, 12.56, 1232, 20.0

The file is present in the bin/debug folder of the sample code provided.

Test case:

```
Employee Details: Jetson  Jordon ID#:1232 Weekly Income: 251.2
Employee Details: Cogswall  James ID#:7165 Weekly Income: 684.725
Employee Details: Spacelly  George ID#:5903 Weekly Income: 968.032
Employee Details: Elroy  Alsison ID#:123 Weekly Income: 116.62
Employee Details: Rosie  Philip ID#:8080 Weekly Income: 281.019

 After Sorting:
Employee Details: Elroy  Alsison ID#:123 Weekly Income: 116.62
Employee Details: Jetson  Jordon ID#:1232 Weekly Income: 251.2
Employee Details: Spacelly  George ID#:5903 Weekly Income: 968.032
Employee Details: Cogswall  James ID#:7165 Weekly Income: 684.725
Employee Details: Rosie  Philip ID#:8080 Weekly Income: 281.019
```

Reference:

StreamReader:

Implements a TextReader that reads characters from a byte stream in a particular encoding. Please check the reference link below before solving the question.

1. https://docs.microsoft.com/en-us/dotnet/api/system.io.streamreader?view=netframework-4.7.2
2. https://www.dotnetperls.com/streamreader