A plane collects data on the height of sea waves at a certain regularity. The plane travels along the same route, collecting the following data:

Date, Pilot name, departure location, and measurement data (10 pieces of data).

The data is collected in a CSV file called **meres.csv**. The file is attached to the assignment.

Create a class called **FileHandler**. The class should have a hidden data member of type string **fileName** and a hidden data member of type bool **fileExists**. For both data members, create a writable and readable property. The class should also have a hidden two-dimensional string array field called **fileArray** (2 scores)

Create a constructor that has an input parameter, the file name, and calls the **FileLoaded** method based on the file name and loads the fileArray array. (2 scores)

Create a method called **FileLoaded** that loads the data in meres.csv. The signature of the method is: **string[,] FileLoaded(string fileName)**. The input parameter can be the name and extension of the file containing the data. The method returns a two-dimensional string array whose rows contain the rows of the file and whose columns contain the columns of the file. (8 scores)

Create a method called **ToWrite** that writes the data of the matrix specified in the parameter to the screen in a way that displays the rows and columns of the matrix in a tabular format. (The lines of the table do not need to be drawn. The signature of the method is: **void ToWrite(string [,] matrix)**. (5 scores.)

Create a **BiggestWave** method that returns the date when the largest wave was measured. The return value of the method is a string, and its input parameter is an array containing the data. The signature of the method is: **string BiggestWave(string [,] matrix)**. (10 scores.)

Create a method that determines whether a measurement was made with these data based on the pilot's name and departure location. The signature of the method is: **bool ItsExist(string [,] matrix, string pilotsName, string departureLocation)** (10 scores.)

Implement a method that returns all measured data rounded up. The method should not have a return value, and its input parameter should be a matrix containing the data. The name of the method should be: **RoundUp**. (10 scores.)

|  |
| --- |
| Testing |
| In the main method, test each program element you have created. Write the results to a console screen large enough to fit one line next to another. Request data from the user for the elements of the IstExists method. (5 scores.) |

Create a **FileWriter** class that creates a .csv file based on the above. The class should have a hidden data member containing the hidden text **fileName**, a bool type **isProblem** data member. and a two-dimensional array of string type called **measurementData**. The fileName data member and the measurementData array should have a writable readable property. (2 scores)

Create a constructor that requests the file name, then uses the **UpLoad** method to load the class's measurementData array, checks with the IsExist method that the file you want to create does not exist, and if it does not exist, creates it, and if it does, sets the **IsProblem** property to true. (1 scores)

Create a method with the signature **bool IsExists()** that decides whether the file specified in fileName exists or not. If the file exists, return true. (2 scores)

Create a method with the signature **string[,] matrix UpLoad()** that returns an uploaded two-dimensional array. The array contains the data in the form meres.csv, so the first row contains the column titles, the other rows contain the Date, Pilot name, departure location, and the 10 measurement data. The method automatically forms the first row of the array in this format:

*Date;Pilot name;Departure location;Data1;Data2;Data3;Data4;Data5;Data6;Data7;Data8;Data9;Data10*

The date data is automatically filled in by always entering the current date. It asks the user for the pilot name and departure location, and fills the measurement data with one-decimal-digit random numbers between 0 and 100. (20 scores)

|  |
| --- |
| Testing |
| In the main method, instantiate an object named file1 and run its constructor with the parameter meres2.csv. Add at least 3 measurements to the object. Save the contents of the array to the specified file.  Do the same with the object created using the parameter file2.csv. |

Create a class called **FileStore** that stores data of type FileHandler in a list. The name of the list should be fileList, visibility: hidden. (2 scores)

Create a constructor that adds a new data item to the fileList list using the method called **AddToList**. (1 scores)

Create a public method **AddToList** that adds the FileHandler object in the parameter to the list. (2 scores)

Create a public **void Run(string fileName)** method that prints the data stored in the FileHandler objects that belong to the given file name, if possible, if not, print to the screen that the file cannot be written because it already exists. To do this, we override the ToString() method of the FileHandler class! (10 scores)

|  |
| --- |
| Testing |
| Implement an object called fileStore that adds the data containing the file meres.csv to the list. Add the objects containing the data file1.csv and file2.csv as well. Ask for a filename and then print its contents to the screen. (3 scores). |

Evaluation:

Assignments with translation-time errors cannot be evaluated (they will receive an insufficient grade). Plagiarized assignments (copied from other students) will result in an insufficient grade for all parties involved. Students who submit a Plagiarized Assignment cannot make up the Zárthely Paper, they can only correct it during the exam period (re-exam).