

```
1  //-- *****
2  //-- CLASS:      ObservationDB
3  //-- AUTHOR:     Paul McKillop
4  //-- CREATED:    19 January 2019
5  //-- PURPOSE:    Retrieve and store Observation data
6  //-- *****
7
8
9  using System;
10 using System.Collections.Generic;
11 using System.Data;
12 using System.IO;
13 using System.Linq;
14 using System.Text;
15 using System.Threading.Tasks;
16
17 namespace ProductPerformancePrototype
18 {
19     public class ObservationDB
20     {
21         /// <summary>
22         /// Write an observation class dataset to the text file
23         /// </summary>
24         /// <param name="myObservation"></param>
25         /// <returns>Bool to indicate success of the write process</returns>
26         public static bool WriteObservation(Observation myObservation)
27         {
28             //-- Get the data into the right format for writing to text file
29             string observationData = myObservation.AllObservationData();
30
31             //-- Use StreamWriter to write the data
32             using (StreamWriter writer = new StreamWriter(@"D:
33                 \observations.txt", true))
34             {
35                 //-- Write the line of data
36                 writer.WriteLine(observationData);
37
38                 //-- Close the StreamWriter
39                 writer.Close();
40                 //-- Dispose of it from memory. Shouldn't be needed
41                 //-- because of the 'using()' method but extra precaution
42                 writer.Dispose();
43             }
44             //-- got this far
45             return true;
46         }
47
48
49
50         //-- Import the data in the text file into a DataTable
51         public static DataTable GetAllObservationsData()
52         {
53             //-- Object to hold the data
54             var dt = new DataTable();
55
```

```
56         //-- Get it
57         dt = ImportData.GetTextFileData(@"D:\observations.txt");
58
59         //-- Return it in the method
60         return dt;
61     }
62
63
64     //-- Get all observations unformatted
65     public static List<string> GetAllObservationsUnformatted()
66     {
67         List<string> tempList = new List<string>();
68
69         //-- object for data
70         var dt = new DataTable();
71
72         //-- Populate
73         dt = ImportData.GetTextFileData(@"D:\observations.txt");
74
75         //-- loop and fill the list
76         foreach (DataRow row in dt.Rows)
77         {
78             var currentObservation = new Observation()
79             {
80                 Person = row.Field<string>(0),
81                 ProductType = row.Field<string>(1),
82                 Product = row.Field<string>(2),
83                 ProductSize = row.Field<string>(3),
84                 FirstUse = row.Field<string>(4),
85                 Continuous = row.Field<string>(5),
86                 Observation1 = row.Field<string>(6),
87                 Observation2 = row.Field<string>(7),
88                 Observation3 = row.Field<string>(8)
89             };
90
91             //-- add
92             tempList.Add(currentObservation.AllObservationData());
93         }
94
95         return tempList;
96     }
97
98
99     public static List<string> GetAllObservationsShort()
100     {
101         List<string> tempList = new List<string>();
102
103         //-- object for data
104         var dt = new DataTable();
105
106         //-- Populate
107         dt = ImportData.GetTextFileData(@"D:\observations.txt");
108
109         //-- loop and fill the list
110         foreach (DataRow row in dt.Rows)
111         {
```

```
112         var currentObservation = new Observation()
113         {
114             Person = row.Field<string>(0),
115             ProductType = row.Field<string>(1),
116             Product = row.Field<string>(2),
117             ProductSize = row.Field<string>(3),
118             FirstUse = row.Field<string>(4),
119             Continuous = row.Field<string>(5),
120             Observation1 = row.Field<string>(6),
121             Observation2 = row.Field<string>(7),
122             Observation3 = row.Field<string>(8)
123         };
124
125         //-- add
126         tempList.Add(currentObservation.ObservationDataShort());
127     }
128
129     return tempList;
130 }
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156     public static List<string> ObservationsFormattedPasting()
157     {
158         List<string> tempList = new List<string>();
159
160         return tempList;
161     }
162
163     public static List<string> ObservationsUnformattedPasting()
164     {
165         List<string> tempList = new List<string>();
166
167         return tempList;
```

```
168         }  
169     }  
170 }  
171
```