```
1  //-- *************************
2  //-- CLASS:     RunningCostCalculator
3  //-- AUTHOR:    Paul McKillop
4  //-- CREATED:   03 January 2019 (Modified class structure)
5  //-  PURPOSE:   Handle data for loan attributes/properties
6  //-- *************************
7
8  using System;
9  using System.Collections.Generic;
10 using System.Linq;
11 using System.Text;
12 using System.Threading.Tasks;
13
14 namespace Motoring
15 {
16     public class RunningCostsCalculator
17     {
18         //-- Insurance
19         //-- Annual
20         #region Insurance costs
21         public static double InsuranceCostAnnual(RunningCost myRunningCost)
22         {
23             //-- return value
24             double cost = 0;
25             string period = myRunningCost.InsurancePeriod;
26
27             switch (period)
28             {
29                 case "Annual":
30                     cost = Convert.ToDouble(myRunningCost.Insurance);
31                     break;
32                 case "Monthly":
33                     cost = Convert.ToDouble(myRunningCost.Insurance * 12);
34                     break;
35                 case "Weekly":
36                     cost = Convert.ToDouble(myRunningCost.Insurance * 52);
37                     break;
38                 //-- switch must have a default return value to avoid endless
                       loop or unreachable code
39                 default:
40                     cost = 0;
41                     break;
42             }
43
44             // --now return the value from the method based on the switch
45             return Math.Round(cost, 2);
46         }
47
48         //-- Monthly
49         public static double InsuranceCostMonthly(RunningCost myRunningCost)
50         {
51             double cost = 0;
52             string period = myRunningCost.InsurancePeriod;
53
54             switch (period)
55             {
```

```
56                  case "Annual":
57                      cost = Convert.ToDouble(myRunningCost.Insurance / 12);
58                      break;
59                  case "Monthly":
60                      cost = Convert.ToDouble(myRunningCost.Insurance);
61                      break;
62                  case "Weekly":
63                      cost = Convert.ToDouble(myRunningCost.Insurance * 52 /
                        12);
64                      break;
65                  default:
66                      cost = 0;
67                      break;
68
69              }
70
71              return Math.Round(cost, 2);
72          }
73
74          //-- Weekly
75          public static double InsuranceCostWeekly(RunningCost myRunningCost)
76          {
77              double cost = 0;
78              string period = myRunningCost.InsurancePeriod;
79
80              switch (period)
81              {
82                  case "Annual":
83                      cost = Convert.ToDouble(myRunningCost.Insurance / 52);
84                      break;
85                  case "Monthly":
86                      cost = Convert.ToDouble(myRunningCost.Insurance * 12 /
                        52);
87                      break;
88                  case "Weekly":
89                      cost = Convert.ToDouble(myRunningCost.Insurance);
90                      break;
91                  default:
92                      cost = 0;
93                      break;
94
95              }
96
97              return Math.Round(cost, 2);
98          }
99          #endregion
100
101         //-- Fuel
102         //-- Annual
103         #region Fuel costs
104
105         public static double FuelCostAnnual(RunningCost myRunningCost)
106         {
107             double cost = 0;
108             string period = myRunningCost.FuelPeriod;
109
```

```
110             switch (period)
111             {
112                 case "Annual":
113                     cost = Convert.ToDouble(myRunningCost.Fuel);
114                     break;
115                 case "Monthly":
116                     cost = Convert.ToDouble(myRunningCost.Fuel * 12);
117                     break;
118                 case "Weekly":
119                     cost = Convert.ToDouble(myRunningCost.Fuel * 52);
120                     break;
121                 default:
122                     cost = 0;
123                     break;
124 
125             }
126 
127             return Math.Round(cost, 2);
128         }
129 
130         //-- Monthly
131         public static double FuelCostMonthly(RunningCost myRunningCost)
132         {
133             double cost = 0;
134             string period = myRunningCost.FuelPeriod;
135 
136             switch (period)
137             {
138                 case "Annual":
139                     cost = Convert.ToDouble(myRunningCost.Fuel / 12);
140                     break;
141                 case "Monthly":
142                     cost = Convert.ToDouble(myRunningCost.Fuel);
143                     break;
144                 case "Weekly":
145                     cost = Convert.ToDouble(myRunningCost.Fuel * 52 / 12);
146                     break;
147                 default:
148                     cost = 0;
149                     break;
150 
151             }
152 
153             return Math.Round(cost, 2);
154         }
155 
156         //-- Weekly
157         public static double FuelCostWeekly(RunningCost myRunningCost)
158         {
159             double cost = 0;
160             string period = myRunningCost.FuelPeriod;
161 
162             switch (period)
163             {
164                 case "Annual":
165                     cost = Convert.ToDouble(myRunningCost.Fuel / 52);
```

```
166                     break;
167                 case "Monthly":
168                     cost = Convert.ToDouble(myRunningCost.Fuel * 12 / 52);
169                     break;
170                 case "Weekly":
171                     cost = Convert.ToDouble(myRunningCost.Fuel);
172                     break;
173                 default:
174                     cost = 0;
175                     break;
176 
177             }
178 
179             return Math.Round(cost, 2);
180         }
181         #endregion
182 
183 
184         //-- Servicing
185         #region Servicing costs
186         public static double ServicingCostAnnual(RunningCost myRunningCost)
187         {
188             double cost = 0;
189             string period = myRunningCost.ServicingPeriod;
190 
191             switch (period)
192             {
193                 case "Annual":
194                     cost = Convert.ToDouble(myRunningCost.Servicing);
195                     break;
196                 case "Monthly":
197                     cost = Convert.ToDouble(myRunningCost.Servicing * 12);
198                     break;
199                 case "Weekly":
200                     cost = Convert.ToDouble(myRunningCost.Servicing * 52);
201                     break;
202                 default:
203                     cost = 0;
204                     break;
205 
206             }
207 
208             return Math.Round(cost, 2);
209         }
210 
211         public static double ServicingCostMonthly(RunningCost myRunningCost)
212         {
213             double cost = 0;
214             string period = myRunningCost.ServicingPeriod;
215 
216             switch (period)
217             {
218                 case "Annual":
219                     cost = Convert.ToDouble(myRunningCost.Servicing / 12);
220                     break;
221                 case "Monthly":
```

```
222                     cost = Convert.ToDouble(myRunningCost.Servicing);
223                     break;
224                 case "Weekly":
225                     cost = Convert.ToDouble(myRunningCost.Servicing * 52 /
                    12);
226                     break;
227                 default:
228                     cost = 0;
229                     break;
230 
231             }
232 
233             return Math.Round(cost, 2);
234         }
235 
236         public static double ServicingCostWeekly(RunningCost myRunningCost)
237         {
238             double cost = 0;
239             string period = myRunningCost.ServicingPeriod;
240 
241             switch (period)
242             {
243                 case "Annual":
244                     cost = Convert.ToDouble(myRunningCost.Servicing / 52);
245                     break;
246                 case "Monthly":
247                     cost = Convert.ToDouble(myRunningCost.Servicing * 12 /
                    52);
248                     break;
249                 case "Weekly":
250                     cost = Convert.ToDouble(myRunningCost);
251                     break;
252                 default:
253                     cost = 0;
254                     break;
255 
256             }
257 
258             return Math.Round(cost, 2);
259         }
260         #endregion
261 
262         //-- RoadTax
263         #region Road Tax Costs
264         public static double RoadTaxAnnual(RunningCost myRunningCost)
265         {
266             double cost = 0;
267             string period = myRunningCost.RoadTaxPeriod;
268 
269             switch (period)
270             {
271                 case "Annual":
272                     cost = Convert.ToDouble(myRunningCost.RoadTax);
273                     break;
274                 case "Monthly":
275                     cost = Convert.ToDouble(myRunningCost.RoadTax * 12);
```

```
276                     break;
277                 case "Weekly":
278                     cost = Convert.ToDouble(myRunningCost.RoadTax * 52);
279                     break;
280                 default:
281                     cost = 0;
282                     break;
283 
284             }
285 
286             return Math.Round(cost, 2);
287         }
288 
289         public static double RoadTaxMonthly(RunningCost myRunningCost)
290         {
291             double cost = 0;
292             string period = myRunningCost.RoadTaxPeriod;
293 
294             switch (period)
295             {
296                 case "Annual":
297                     cost = Convert.ToDouble(myRunningCost.RoadTax / 12);
298                     break;
299                 case "Monthly":
300                     cost = Convert.ToDouble(myRunningCost.RoadTax);
301                     break;
302                 case "Weekly":
303                     cost = Convert.ToDouble(myRunningCost.RoadTax * 52 / 12);
304                     break;
305                 default:
306                     cost = 0;
307                     break;
308 
309             }
310 
311             return Math.Round(cost, 2);
312         }
313 
314         public static double RoadTaxWeekly(RunningCost myRunningCost)
315         {
316             double cost = 0;
317             string period = myRunningCost.RoadTaxPeriod;
318 
319             switch (period)
320             {
321                 case "Annual":
322                     cost = Convert.ToDouble(myRunningCost.RoadTax / 52);
323                     break;
324                 case "Monthly":
325                     cost = Convert.ToDouble(myRunningCost.RoadTax * 12 / 52);
326                     break;
327                 case "Weekly":
328                     cost = Convert.ToDouble(myRunningCost.RoadTax);
329                     break;
330                 default:
331                     cost = 0;
```

```
332                         break;
333
334             }
335
336             return Math.Round(cost, 2);
337         }
338         #endregion
339
340
341         //-- Total Weekly Running costs
342         public static double TotalWeeklyRunningCost(RunningCost myRunningCost)
343         {
344             double totalWeeklyCost = 0;
345             totalWeeklyCost += InsuranceCostWeekly(myRunningCost);
346             totalWeeklyCost += FuelCostWeekly(myRunningCost);
347             totalWeeklyCost += ServicingCostWeekly(myRunningCost);
348             totalWeeklyCost += RoadTaxWeekly(myRunningCost);
349
350             return totalWeeklyCost;
351         }
352
353
354
355         //-- Template
356         #region Method Template
357         public static double CalculatorTemplate(RunningCost myRunningCost)
358         {
359             double cost = 0;
360             string period = "Z";
361
362             switch (period)
363             {
364                 case "Annual":
365                     cost = Convert.ToDouble(0);
366                     break;
367                 case "Monthly":
368                     cost = Convert.ToDouble(0);
369                     break;
370                 case "Weekly":
371                     cost = Convert.ToDouble(0);
372                     break;
373                 default:
374                     cost = 0;
375                     break;
376
377             }
378
379             return Math.Round(cost, 2);
380         }
381         #endregion
382     }
383 }
384
```