```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using System.Windows;
7  using System.Windows.Controls;
8  using System.Windows.Data;
9  using System.Windows.Documents;
10 using System.Windows.Input;
11 using System.Windows.Media;
12 using System.Windows.Media.Animation;
13 using System.Windows.Media.Imaging;
14 using System.Windows.Navigation;
15 using System.Windows.Shapes;
16
17 namespace Motoring
18 {
19     /// <summary>
20     /// Interaction logic for PageSummary.xaml
21     /// </summary>
22     public partial class PageSummary : Page
23     {
24         //-- manage the data module wide
25         CostSummary summary = new CostSummary();
26         Loan loan = new Loan();
27         RunningCost runningCost = new RunningCost();
28
29         //-- Summary value variables
30         private decimal totalLoanCost = 0;
31         private double totalRunningCost = 0;
32         private double totalCostOfOwnership = 0;
33
34
35         //-- Period list for combo
36         //-- Combo items list of strings
37         List<string> costPeriods = new List<string>();
38
39         //-- Budget period variables
40         private double periodLoan = 0;
41         private double periodRunningCost = 0;
42         private double periodTotalCost = 0;
43
44         private string periodSelected = "Annual";    //-- Initialised toi
             Annual
45
46
47         public PageSummary(CostSummary summaryPassed)
48         {
49             InitializeComponent();
50             summary = summaryPassed;
51             loan = summary.CurrentLoan;
52             runningCost = summary.RunningCost;
53
54             costPeriods = Periods();
55
```

```
56              BudgetPeriodCombo.ItemsSource = costPeriods;
57
58              //--- Show values held
59              MessageBox.Show(DataValuesSummary());
60
61              //-- Calulate and load total costs
62              ShowTotalCosts();
63          }
64
65          private void LoanPageButton_OnClick(object sender, RoutedEventArgs e)
66          {
67              var pageLoan = new PageLoan();
68              this.NavigationService.Navigate(pageLoan);
69          }
70          // -- Combo box data population
71          List<string> Periods()
72          {
73              List<string> myList = new List<string>
74              {
75                  "Annual",
76                  "Monthly",
77                  "Weekly"
78              };
79
80              return myList;
81          }
82
83          private void BudgetPeriodCombo_OnLoaded(object sender, RoutedEventArgs
                e)
84          {
85              var combo = sender as ComboBox;
86              combo.ItemsSource = costPeriods;
87              combo.SelectedIndex = 0;
88          }
89
90          private void BudgetPeriodCombo_OnSelectionChanged(object sender,
              SelectionChangedEventArgs e)
91          {
92              var selectedComboItem = sender as ComboBox;
93              periodSelected = selectedComboItem.SelectedItem as string;
94
95              //-- work out periodLoan value
96              switch (periodSelected)
97              {
98                  case "Annual":
99                      periodLoan = Convert.ToDouble
                        (LoanCalculator.LoanAnnualPayment(loan));
100                     break;
101                 case "Monthly":
102                     periodLoan = Convert.ToDouble
                        (LoanCalculator.LoanMonthlyPayment(loan));
103                     break;
104                 case "Weekly":
105                     periodLoan = Convert.ToDouble
                        (LoanCalculator.LoanWeeklyPayment(loan));
106                     break;
```

```
107             }
108
109             //-- Period running costs
110             switch (periodSelected)
111             {
112                 case "Annual":
113                     periodRunningCost =
                        RunningCostsCalculator.TotalWeeklyRunningCost(runningCost)
                         * 52;
114                     break;
115                 case "Monthly":
116                     periodRunningCost =
                        RunningCostsCalculator.TotalWeeklyRunningCost(runningCost)
                         * 52 / 12;
117                     break;
118                 case "Weekly":
119                     periodRunningCost =
                        RunningCostsCalculator.TotalWeeklyRunningCost
                        (runningCost);
120                     break;
121             }
122
123             periodTotalCost = periodLoan + periodRunningCost;
124
125             //-- Put values into the text blocks
126
127             LoanPeriodTextBlock.Text = periodLoan.ToString("C");
128             RunningCostPeriodTextBlock.Text = periodRunningCost.ToString("C");
129
130             //-- Total of period costs
131             TotalPeriodCostsTextBlock.Text = periodTotalCost.ToString("C");
132         }
133
134         private void ShowTotalCosts()
135         {
136             totalLoanCost = LoanCalculator.LoanTotalPayment(loan);
137             TotalLoanTextBlock.Text = totalLoanCost.ToString("C");
138
139             //-- Total running costs
140             totalRunningCost = RunningCostsCalculator.TotalWeeklyRunningCost
                (runningCost) * 52 * loan.LoanTermYears;
141             TotalRunnincCostTextBlock.Text = totalRunningCost.ToString("C");
142
143             //-- Total Cost of ownership
144             double loanCost = Convert.ToDouble(totalLoanCost);
145             totalCostOfOwnership = (loanCost + totalRunningCost +
                loan.CarDeposit);
146
147             TotalOwenershipTextBlock.Text = totalCostOfOwnership.ToString
                ("C");
148         }
149
150         string DataValuesSummary()
151         {
152             string message = "";
153
```

```
154             var valuesHeld = new StringBuilder();
155
156             //-- Loan
157             valuesHeld.Append("Car price: ").Append(loan.CarPrice.ToString
                  ()).AppendLine();
158             valuesHeld.Append("Deposit: ").Append(loan.CarDeposit.ToString
                  ()).AppendLine();
159             valuesHeld.Append("Loan term: ").Append
                  (loan.LoanTermYears.ToString()).AppendLine();
160             valuesHeld.Append("Interest rate: ").Append(loan.LoanRate.ToString
                  ()).AppendLine();
161             //-- Running Costs
162             valuesHeld.Append("Insurance cost: ").Append
                  (runningCost.Insurance.ToString()).AppendLine();
163             valuesHeld.Append("Insurance period: ").Append
                  (runningCost.InsurancePeriod).AppendLine();
164             valuesHeld.Append("Fuel cost: ").Append(runningCost.Fuel.ToString
                  ()).AppendLine();
165             valuesHeld.Append("Fuel period: ").Append
                  (runningCost.FuelPeriod).AppendLine();
166             valuesHeld.Append("Servicing cost: ").Append
                  (runningCost.Servicing.ToString()).AppendLine();
167             valuesHeld.Append("Servicing period: ").Append
                  (runningCost.ServicingPeriod).AppendLine();
168             valuesHeld.Append("Road Tax cost: ").Append
                  (runningCost.RoadTax.ToString()).AppendLine();
169             valuesHeld.Append("Road Tax period: ").Append
                  (runningCost.RoadTaxPeriod).AppendLine();
170
171             //-- convert StringBuilder to string and return
172             message = valuesHeld.ToString();
173
174             return message;
175
176         }
177     }
178 }
179
```