

```
1 using System;
2 using System.Collections.Generic;
3 using System.Text;
4 using System.Windows;
5 using System.Windows.Controls;
6
7
8 /* TITLE:      PageSummary
9  * AUTHOR:     Paul McKillop
10 * DATE:       October
11 * PURPOSE:    Page interaction code
12 */
13
14 namespace McKillopMotoring
15 {
16     /// <summary>
17     /// Interaction logic for PageSummary.xaml
18     /// </summary>
19     public partial class PageSummary : Page
20     {
21         //-- manage the data module wide
22         CostSummary summary = new CostSummary();
23         Loan loan = new Loan();
24         RunningCost runningCost = new RunningCost();
25
26         //-- Summary value variables
27         private decimal totalLoanCost = 0;
28         private double totalRunningCost = 0;
29         private double totalCostOfOwnership = 0;
30
31         //-- Period list for combo
32         //-- Combo items list of strings
33         List<string> costPeriods = new List<string>();
34
35         //-- Budget period variables
36         private double periodLoan = 0;
37         private double periodRunningCost = 0;
38         private double periodTotalCost = 0;
39
40         private string periodSelected = "Annual";
41
42         public PageSummary(CostSummary summaryPassed)
43         {
44             InitializeComponent();
45             summary = summaryPassed;
46             loan = summary.CurrentLoan;
47             runningCost = summary.RunningCost;
48
49             costPeriods = Periods();
50
51             BudgetPeriodCombo.ItemsSource = costPeriods;
52
53             // -- TODO :: Show values held
```

```
54     MessageBox.Show(DataValuesSummary());
55
56     // -- TODO :: Calculate and show totals
57     ShowTotalCosts();
58     // -- DEBUG REMOVE AT FINAL BUILD
59     //string price =
60     summaryPassed.CurrentLoan.CarPrice.ToString();
61     //MessageBox.Show("The Car Price passed from PageLoan was "
62     + price);
63
64     //string insurance =
65     summaryPassed.RunningCost.Insurance.ToString();
66     //MessageBox.Show("The Insurance cost passed was " +
67     insurance);
68 }
69
70 private void BudgetPeriodCombo_OnLoaded(object sender,
71     RoutedEventArgs e)
72 {
73     var combo = sender as ComboBox;
74     combo.ItemsSource = costPeriods;
75     combo.SelectedIndex = 0;
76 }
77
78 private void BudgetPeriodCombo_OnSelectionChanged(object
79     sender, SelectionChangedEventArgs e)
80 {
81     var selectedComboItem = sender as ComboBox;
82     periodSelected = selectedComboItem.SelectedItem as string;
83
84     //-- work out periodLoan value
85     switch (periodSelected)
86     {
87     case "Annual":
88         periodLoan = Convert.ToDouble(value:
89             LoanCalculator.LoanAnnualPayment(myLoan: loan));
90         break;
91     case "Monthly":
92         periodLoan = Convert.ToDouble(value:
93             LoanCalculator.LoanMonthlyPayment(myLoan: loan));
94         break;
95     case "Weekly":
96         periodLoan = Convert.ToDouble(value:
97             LoanCalculator.LoanWeeklyPayment(myLoan: loan));
98         break;
99     }
100
101     //-- Period running costs
102     switch (periodSelected)
103     {
104     case "Annual":
105         periodRunningCost =
106             RunningCostsCalculator.TotalWeeklyRunningCost
```

```

        (myRunningCost: runningCost) * 52;
97         break;
98     case "Monthly":
99         periodRunningCost =
            RunningCostsCalculator.TotalWeeklyRunningCost
            (myRunningCost: runningCost) * 52 / 12;
100        break;
101    case "Weekly":
102        periodRunningCost =
            RunningCostsCalculator.TotalWeeklyRunningCost
            (myRunningCost: runningCost);
103        break;
104    }
105
106    periodTotalCost = periodLoan + periodRunningCost;
107
108    //-- Put values into the text blocks
109
110    LoanPeriodTextBlock.Text = periodLoan.ToString(format: "C");
111    RunningCostPeriodTextBlock.Text =
        periodRunningCost.ToString(format: "C");
112
113    //-- Total of period costs
114    TotalPeriodCostsTextBlock.Text = periodTotalCost.ToString
        (format: "C");
115    }
116
117    private void LoanPageButton_OnClick(object sender,
        RoutedEventArgs e)
118    {
119        var pageLoan = new PageLoan();
120        this.NavigationService.Navigate(root: pageLoan);
121    }
122
123    // - Utility
124    // -- Combo box data population
125    List<string> Periods()
126    {
127        List<string> myList = new List<string>
128        {
129            "Annual",
130            "Monthly",
131            "Weekly"
132        };
133
134        return myList;
135    }
136
137    private void ShowTotalCosts()
138    {
139        totalLoanCost = LoanCalculator.LoanTotalPayment(myLoan:
        loan);
140    }

```

```
142
143     //-- Total running costs
144     totalRunningCost =
145         RunningCostsCalculator.TotalWeeklyRunningCost(myRunningCost:
146             runningCost) * 52 * loan.LoanTermYears;
147     TotalRunningCostTextBlock.Text = totalRunningCost.ToString
148         (format: "C");
149
150     //-- Total Cost of ownership
151     double loanCost = Convert.ToDouble(value: totalLoanCost);
152     totalCostOfOwnership = (loanCost + totalRunningCost +
153         loan.CarDeposit);
154
155     TotalOwnershipTextBlock.Text =
156         totalCostOfOwnership.ToString(format: "C");
157 }
158
159 string DataValuesSummary()
160 {
161     string message = "";
162
163     var valuesHeld = new StringBuilder();
164
165     //-- Loan
166     valuesHeld.Append(value: "Car price: ").Append(value:
167         loan.CarPrice.ToString()).AppendLine();
168     valuesHeld.Append(value: "Deposit: ").Append(value:
169         loan.CarDeposit.ToString()).AppendLine();
170     valuesHeld.Append(value: "Loan term: ").Append(value:
171         loan.LoanTermYears.ToString()).AppendLine();
172     valuesHeld.Append(value: "Interest rate: ").Append(value:
173         loan.LoanRate.ToString()).AppendLine();
174     //-- Running Costs
175     valuesHeld.Append(value: "Insurance cost: ").Append(value:
176         runningCost.Insurance.ToString()).AppendLine();
177     valuesHeld.Append(value: "Insurance period: ").Append(value:
178         runningCost.InsurancePeriod).AppendLine();
179     valuesHeld.Append(value: "Fuel cost: ").Append(value:
180         runningCost.Fuel.ToString()).AppendLine();
181     valuesHeld.Append(value: "Fuel period: ").Append(value:
182         runningCost.FuelPeriod).AppendLine();
183     valuesHeld.Append(value: "Servicing cost: ").Append(value:
184         runningCost.Servicing.ToString()).AppendLine();
185     valuesHeld.Append(value: "Servicing period: ").Append(value:
186         runningCost.ServicingPeriod).AppendLine();
187     valuesHeld.Append(value: "Road Tax cost: ").Append(value:
188         runningCost.RoadTax.ToString()).AppendLine();
189     valuesHeld.Append(value: "Road Tax period: ").Append(value:
190         runningCost.RoadTaxPeriod).AppendLine();
191
192     //-- convert StringBuilder to string and return
193     message = valuesHeld.ToString();
194 }
```

```
178         return message;
179
180     }
181
182
183     }
184 }
185
```