

```
DoubleProperty pVal = new SimpleDoubleProperty(periodsVal)
```

```
// The actual period is gotten as follows
DoubleProperty dp = new SimpleDoubleProperty();
dp.bind(pVal.divide(365));

// Now we can perform the calculations
DoubleProperty ie = new SimpleDoubleProperty();
ie.bind(principal.multiply(interestRate.multiply(dp)));

DoubleProperty df = new SimpleDoubleProperty();
df.bind(principal.add(ie));

// Display the values
txtInterestEarned.setText("$" + truncate(ie.getValue()));
txtFutureValue.setText("$" + truncate(df.getValue()));
}
```

Simple Interest

Loan Preparation

Principal:

6425.00

Interest Rate:

14.65 %

Loan Start Date:

4/10/2006

Loan End Date:

11/8/2007

Periods:

577 day

Results

Interest Earned:

\$1487.97

Future Value:

\$7912.97

Close

```

1 <?xml version="1.0" encoding="UTF-8"?>
2
3 <?import javafx.geometry.Insets?>
4 <?import javafx.scene.control.Button?>
5 <?import javafx.scene.control.DatePicker?>
6 <?import javafx.scene.control.Label?>
7 <?import javafx.scene.control.TextField?>
8 <?import javafx.scene.layout.ColumnConstraints?>
9 <?import javafx.scene.layout.GridPane?>
10 <?import javafx.scene.layout.HBox?>
11 <?import javafx.scene.layout.RowConstraints?>
12 <?import javafx.scene.layout.StackPane?>
13 <?import javafx.scene.layout.VBox?>
14
15 <VBox maxHeight="500.0" maxWidth="600.0" minHeight="-Infinity"
    minWidth="-Infinity" prefHeight="350.0" prefWidth="385.0"
    spacing="20.0" style="-fx-background-color: #fff;" xmlns="http
    ://javafx.com/javafx/17" xmlns:fx="http://javafx.com/fxml/1"
    fx:controller="com.example.vphw04.HelloController">
16     <padding>
17         <Insets bottom="20.0" left="20.0" right="20.0" top="20.0"
18     />
19     </padding>
20     <children>
21         <StackPane maxHeight="200.0" maxWidth="345.0" prefHeight
22     ="257.0" prefWidth="395.0">
23             <children>
24                 <HBox alignment="CENTER" maxHeight="200.0"
25     maxWidth="345.0" prefHeight="228.0" prefWidth="360.0" style="-
26     fx-border-color: gray;">
27                     <children>
28                         <GridPane hgap="5.0" maxHeight="200.0"
29     prefHeight="210.0" prefWidth="318.0">
30                             <columnConstraints>
31                                 <ColumnConstraints hgrow="SOMETIMES"
32     maxWidth="120.0" minWidth="10.0" prefWidth="120.0" />
33                                 <ColumnConstraints hgrow="SOMETIMES"
34     minWidth="10.0" prefWidth="100.0" />
35                                 <ColumnConstraints hgrow="SOMETIMES"
36     minWidth="10.0" prefWidth="100.0" />
37                                 <ColumnConstraints hgrow="SOMETIMES"
38     minWidth="10.0" prefWidth="100.0" />
39                             </columnConstraints>

```

```

31         <rowConstraints>
32             <RowConstraints maxHeight="40.0"
minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
33             <RowConstraints maxHeight="40.0"
minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
34             <RowConstraints maxHeight="40.0"
minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
35             <RowConstraints maxHeight="40.0"
minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
36             <RowConstraints maxHeight="40.0"
minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
37         </rowConstraints>
38         <HBox.margin>
39             <Insets bottom="10.0" left="20.0"
right="20.0" top="10.0" />
40         </HBox.margin>
41         <children>
42             <DatePicker fx:id="dtpStartDate"
prefHeight="25.0" prefWidth="244.0" GridPane.columnIndex="1"
GridPane.columnSpan="3" GridPane.rowIndex="2" />
43             <Label text="Periods:" GridPane.
rowIndex="4" />
44             <Label text="Principal:" />
45             <Label text="Loan End Date:" GridPane.
rowIndex="3" />
46             <Label text="Loan Start Date:"
GridPane.rowIndex="2" />
47             <Label text="Interest Rate:" GridPane.
rowIndex="1" />
48             <Label text="\%" GridPane.columnIndex
="2" GridPane.rowIndex="1" />
49             <Label text="day" GridPane.columnIndex
="2" GridPane.rowIndex="4" />
50             <TextField fx:id="txtPrincipal"
alignment="CENTER_RIGHT" text="0.00" GridPane.columnIndex="1"
/>
51             <TextField fx:id="txtInterestRate"
alignment="CENTER_RIGHT" text="0.00" GridPane.columnIndex="1"
GridPane.rowIndex="1" />
52             <DatePicker fx:id="dtpLoanEndDate"
prefHeight="25.0" prefWidth="238.0" GridPane.columnIndex="1"
GridPane.columnSpan="3" GridPane.rowIndex="3" />
53             <TextField fx:id="txtPeriods"

```

```

53 alignment="CENTER_RIGHT" text="1" GridPane.columnIndex="1"
  GridPane.rowIndex="4" />
54         <Button fx:id="btnCalculate"
mnemonicParsing="false" prefHeight="25.0" prefWidth="83.0"
text="Calculate" visible="false" GridPane.columnIndex="3"
GridPane.rowIndex="4" />
55     </children>
56 </GridPane>
57 </children></HBox>
58     <Label alignment="CENTER" prefHeight="17.0"
prefWidth="100.0" style="-fx-background-color: #fff;" text="
Loan Preparation" translateX="-120.0" translateY="-100.0" />
59 </children>
60 </StackPane>
61 <StackPane maxWidth="345.0" prefHeight="54.0" prefWidth
="410.0">
62     <children>
63         <HBox alignment="CENTER" maxWidth="345.0"
prefHeight="100.0" prefWidth="200.0" style="-fx-border-color:
gray;">
64             <children>
65                 <GridPane alignment="CENTER" hgap="5.0">
66                     <columnConstraints>
67                         <ColumnConstraints hgrow="SOMETIMES"
maxWidth="150.0" minWidth="10.0" prefWidth="105.0" />
68                         <ColumnConstraints hgrow="SOMETIMES"
maxWidth="100.0" minWidth="10.0" prefWidth="85.0" />
69                         <ColumnConstraints hgrow="SOMETIMES"
maxWidth="100.0" minWidth="10.0" prefWidth="90.0" />
70                         <ColumnConstraints hgrow="SOMETIMES"
maxWidth="100.0" minWidth="10.0" prefWidth="85.0" />
71                     </columnConstraints>
72                     <rowConstraints>
73                         <RowConstraints minHeight="10.0"
prefHeight="30.0" vgrow="SOMETIMES" />
74                     </rowConstraints>
75                     <children>
76                         <Label prefHeight="17.0" prefWidth="83
.0" text="Interest Earned:" />
77                         <Label text="Future Value:" GridPane.
columnIndex="2" />
78                         <TextField fx:id="txtInterestEarned"
alignment="CENTER_RIGHT" editable="false" text="\$0.00"

```

```
78 GridPane.columnIndex="1" />
79         <TextField fx:id="txtFutureValue"
alignment="CENTER_RIGHT" editable="false" text="\$0.00"
GridPane.columnIndex="3" />
80         </children>
81         <HBox.margin>
82             <Insets bottom="10.0" left="20.0"
right="20.0" top="10.0" />
83         </HBox.margin>
84     </GridPane>
85 </children>
86 </HBox>
87     <Label alignment="CENTER" prefHeight="17.0"
prefWidth="50.0" style="-fx-background-color: #fff;" text="
Results" translateX="-145.0" translateY="-27.0" />
88 </children>
89 </StackPane>
90 <HBox alignment="BOTTOM_RIGHT" maxWidth="345.0"
prefHeight="21.0" prefWidth="410.0">
91     <children>
92         <Button fx:id="btnClose" alignment="CENTER"
contentDisplay="CENTER" mnemonicParsing="false" onAction="#
close" prefHeight="25.0" prefWidth="108.0" text="Close" />
93     </children>
94 </HBox>
95 </children>
96 </VBox>
97
```

```
1 package com.example.vphw04;
2
3 import javafx.application.Platform;
4 import javafx.beans.InvalidationListener;
5 import javafx.beans.property.*;
6 import javafx.beans.value.ChangeListener;
7 import javafx.beans.value.ObservableValue;
8 import javafx.event.ActionEvent;
9 import javafx.fxml.FXML;
10 import javafx.scene.control.Button;
11 import javafx.scene.control.DatePicker;
12 import javafx.scene.control.TextField;
13
14 import java.math.RoundingMode;
15 import java.text.DecimalFormat;
16 import java.time.Instant;
17 import java.time.LocalDate;
18 import java.time.ZoneId;
19 import java.util.Date;
20 import java.util.concurrent.TimeUnit;
21
22 public class HelloController {
23
24     @FXML
25     private Button btnClose;
26
27     @FXML
28     private DatePicker dtpLoanEndDate;
29
30     @FXML
31     private DatePicker dtpStartDate;
32
33     @FXML
34     private TextField txtFutureValue;
35
36     @FXML
37     private TextField txtInterestEarned;
38
39     @FXML
40     private TextField txtInterestRate;
41
42     @FXML
43     private TextField txtPeriods;
```

```

44
45     @FXML
46     private TextField txtPrincipal;
47
48     @FXML
49     void initialize() {
50         txtPrincipal.textProperty().addListener(new
ChangeListener<String>() {
51             @Override
52             public void changed(final ObservableValue<?
extends String> observable, final String oldValue, final
String newValue) {
53                 calculate();
54             }
55         });
56         txtInterestRate.textProperty().addListener(new
ChangeListener<String>() {
57             @Override
58             public void changed(final ObservableValue<?
extends String> observable, final String oldValue, final
String newValue) {
59                 calculate();
60             }
61         });
62         txtPeriods.textProperty().addListener(new
ChangeListener<String>() {
63             @Override
64             public void changed(final ObservableValue<?
extends String> observable, final String oldValue, final
String newValue) {
65                 calculate();
66             }
67         });
68
69         dtpStartDate.setValue(LocalDate.now());
70         dtpLoanEndDate.setValue(LocalDate.now());
71
72         dtpStartDate.valueProperty().addListener(new
ChangeListener<LocalDate>() {
73             @Override
74             public void changed(ObservableValue<? extends
LocalDate> observable, LocalDate oldValue, LocalDate newValue
) {

```

```

75         calculate();
76     }
77 });
78
79     dtpLoanEndDate.valueProperty().addListener(new
ChangeListener<LocalDate>() {
80         @Override
81         public void changed(ObservableValue<? extends
LocalDate> observable, LocalDate oldValue, LocalDate newValue
) {
82             calculate();
83         }
84     });
85 }
86
87 void calculate(){
88     double dPrincipal = 0.000, dInterestRate = 0.000,
89         dPeriods = 0.000;
90     double dInterestEarned, dFutureValue;
91     double periodsValue = 0;
92     Date startDate, endDate;
93
94     //Get the value of principal
95     try {
96         dPrincipal = Double.parseDouble(txtPrincipal.
getText().trim());
97         txtPrincipal.setStyle("-fx-border-color: #d3d3d3
; -fx-border-radius: 3px;");
98     } catch (NumberFormatException err) {
99         txtPrincipal.setStyle("-fx-border-color: #f00; -
fx-border-radius: 3px;");
100         return;
101     }
102
103     // Get the interest rate
104     try {
105         dInterestRate = Double.parseDouble(
txtInterestRate.getText().trim()) / 100;
106         txtInterestRate.setStyle("-fx-border-color: #
d3d3d3; -fx-border-radius: 3px;");
107     } catch (NumberFormatException err) {
108         txtInterestRate.setStyle("-fx-border-color: #f00
; -fx-border-radius: 3px;");

```



```

109         return;
110     }
111
112     DoubleProperty principal = new SimpleDoubleProperty(
dPrincipal);
113     DoubleProperty interestRate = new
SimpleDoubleProperty(dInterestRate);
114
115     // Get the start and end dates
116     LocalDate localDateStart = dtpStartDate.getValue();
117     Instant instantStart = Instant.from(localDateStart.
atStartOfDay(ZoneId.systemDefault()));
118     startDate = Date.from(instantStart);
119
120     LocalDate localDateEnd = dtpLoanEndDate.getValue();
121     Instant instantEnd = Instant.from(localDateEnd.
atStartOfDay(ZoneId.systemDefault()));
122     endDate = Date.from(instantEnd);
123
124     // Make sure the end date doesn't occur before the
start date
125     if (startDate.after(endDate)){
126         dtpLoanEndDate.setStyle("-fx-border-color: #f00
; -fx-border-radius: 3px;");
127         dtpLoanEndDate.setValue(dtpStartDate.getValue());
128         return;
129     }else{
130         dtpLoanEndDate.setStyle("-fx-border-color: #
d3d3d3; -fx-border-radius: 3px;");
131     }
132
133     // Get the difference in days
134     // will be the periods
135     long diffInMillies = Math.abs(endDate.getTime() -
startDate.getTime());
136     long days = TimeUnit.DAYS.convert(diffInMillies,
TimeUnit.MILLISECONDS);
137     if(days == 0) days = 1;
138     txtPeriods.setText(Long.toString(days));
139
140     // Because we will allow the user to directly specify
141     // the number of days, let's get the period from its
text box

```

```

142         try {
143             periodsValue = Double.parseDouble(txtPeriods.
getText().trim());
144             txtPeriods.setStyle("-fx-border-color: #d3d3d3; -
fx-border-radius: 3px;");
145         } catch (NumberFormatException err) {
146             txtPeriods.setStyle("-fx-border-color: #f00; -fx-
border-radius: 3px;");
147             return;
148         }
149
150         DoubleProperty pVal = new SimpleDoubleProperty(
periodsValue);
151
152         // The actual period is gotten as follows
153         DoubleProperty dp = new SimpleDoubleProperty();
154         dp.bind(pVal.divide(365));
155
156         // Now we can perform the calculations
157         DoubleProperty ie = new SimpleDoubleProperty();
158         ie.bind(principal.multiply(interestRate.multiply(dp
)));
159
160         DoubleProperty df = new SimpleDoubleProperty();
161         df.bind(principal.add(ie));
162
163         // Display the values
164         txtInterestEarned.setText("$" + truncate(ie.getValue
()));
165         txtFutureValue.setText("$" + truncate(df.getValue
()));
166     }
167
168     @FXML
169     void close(ActionEvent event) {
170         Platform.exit();
171     }
172
173     public static String truncate(double input) {
174         DecimalFormat decimalFormat = new DecimalFormat("0.00
");
175         decimalFormat.setRoundingMode(RoundingMode.HALF_UP);
176         String formatResult = decimalFormat.format(input);

```

```
177         return formatResult.trim();  
178     }  
179 }  
180
```