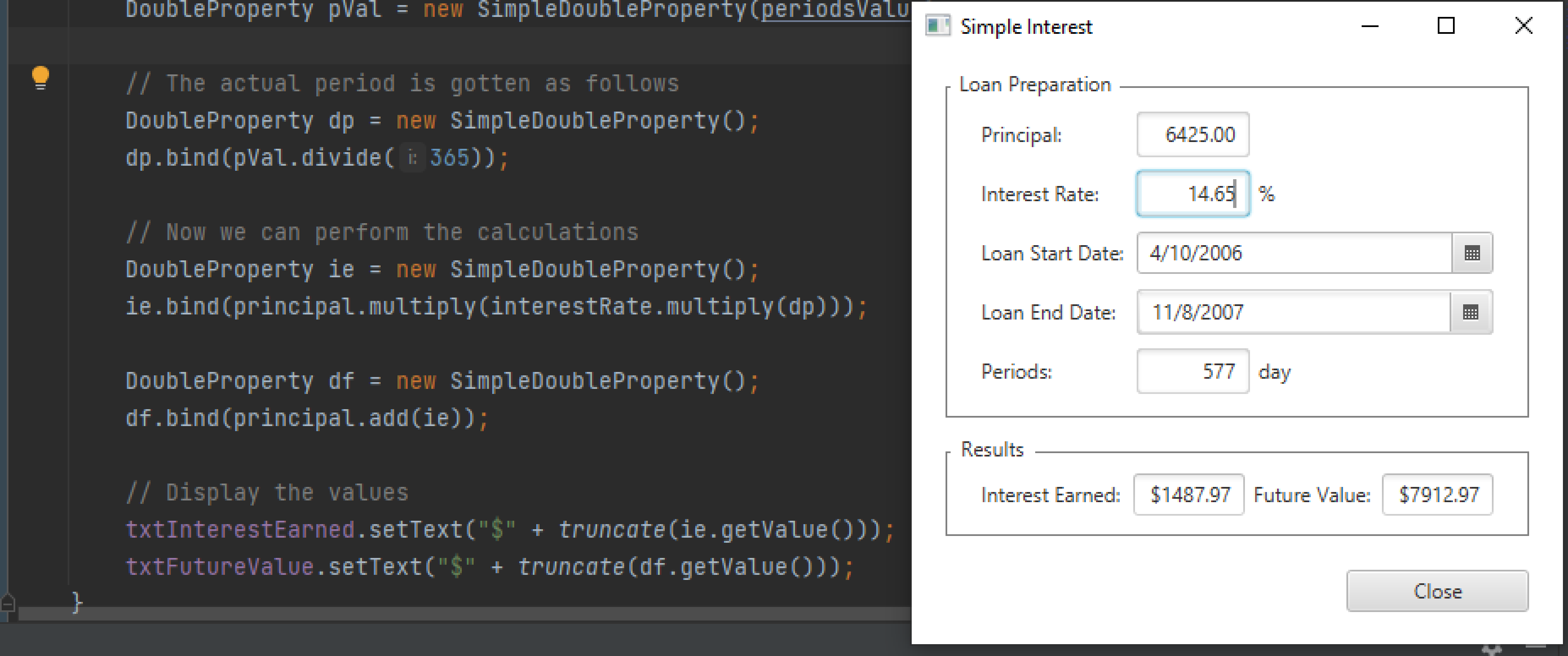

```
        DoubleProperty pVal = new SimpleDoubleProperty(periodsValu

        // 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()));
    }
```

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

```
<?xml version="1.0" encoding="UTF-8"?>

<?import javafx.geometry.Insets?>
<?import javafx.scene.control.Button?>
<?import javafx.scene.control.DatePicker?>
<?import javafx.scene.control.Label?>
<?import javafx.scene.control.TextField?>
<?import javafx.scene.layout.ColumnConstraints?>
<?import javafx.scene.layout.GridPane?>
<?import javafx.scene.layout.HBox?>
<?import javafx.scene.layout.RowConstraints?>
<?import javafx.scene.layout.StackPane?>
<?import javafx.scene.layout.VBox?>

<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">
   <padding>
      <Insets bottom="20.0" left="20.0" right="20.0" top="20.0" />
   </padding>
   <children>
      <StackPane maxHeight="200.0" maxWidth="345.0" prefHeight="257.0" prefWidth="395.0">
         <children>
            <HBox alignment="CENTER" maxHeight="200.0" maxWidth="345.0" prefHeight="228.0" prefWidth="360.0" style="-fx-border-color: gray;">
               <children>
                  <GridPane hgap="5.0" maxHeight="200.0" prefHeight="210.0" prefWidth="318.0">
                    <columnConstraints>
                      <ColumnConstraints hgrow="SOMETIMES" maxWidth="120.0" minWidth="10.0" prefWidth="120.0" />
                        <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                        <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                      <ColumnConstraints hgrow="SOMETIMES" minWidth="10.0" prefWidth="100.0" />
                    </columnConstraints>
                    <rowConstraints>
                      <RowConstraints maxHeight="40.0" minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                      <RowConstraints maxHeight="40.0" minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                      <RowConstraints maxHeight="40.0" minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                        <RowConstraints maxHeight="40.0" minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                        <RowConstraints maxHeight="40.0" minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                    </rowConstraints>
                     <HBox.margin>
                        <Insets bottom="10.0" left="20.0" right="20.0" top="10.0" />
                     </HBox.margin>
                     <children>
                        <DatePicker fx:id="dtpStartDate" prefHeight="25.0" prefWidth="244.0" GridPane.columnIndex="1" GridPane.columnSpan="3" GridPane.rowIndex="2" />
                        <Label text="Periods:" GridPane.rowIndex="4" />
                        <Label text="Principal:" />
                        <Label text="Loan End Date:" GridPane.rowIndex="3" />
                        <Label text="Loan Start Date:" GridPane.rowIndex="2" />
                        <Label text="Interest Rate:" GridPane.rowIndex="1" />
                        <Label text="\%" GridPane.columnIndex="2" GridPane.rowIndex="1" />
                        <Label text="day" GridPane.columnIndex="2" GridPane.rowIndex="4" />
                        <TextField fx:id="txtPrincipal" alignment="CENTER_RIGHT" text="0.00" GridPane.columnIndex="1" />
                        <TextField fx:id="txtInterestRate" alignment="CENTER_RIGHT" text="0.00" GridPane.columnIndex="1" GridPane.rowIndex="1" />
                        <DatePicker fx:id="dtpLoanEndDate" prefHeight="25.0" prefWidth="238.0" GridPane.columnIndex="1" GridPane.columnSpan="3" GridPane.rowIndex="3" />
                        <TextField fx:id="txtPeriods"
```

```
alignment="CENTER_RIGHT" text="1" GridPane.columnIndex="1" GridPane.rowIndex="4" />
                        <Button fx:id="btnCalculate" mnemonicParsing="false" prefHeight="25.0" prefWidth="83.0" text="Calculate" visible="false" GridPane.columnIndex="3" GridPane.rowIndex="4" />
                     </children>
                  </GridPane>
               </children></HBox>
            <Label alignment="CENTER" prefHeight="17.0" prefWidth="100.0" style="-fx-background-color: #fff;" text="Loan Preparation" translateX="-120.0" translateY="-100.0" />
         </children>
      </StackPane>
      <StackPane maxWidth="345.0" prefHeight="54.0" prefWidth="410.0">
         <children>
            <HBox alignment="CENTER" maxWidth="345.0" prefHeight="100.0" prefWidth="200.0" style="-fx-border-color: gray;">
               <children>
                  <GridPane alignment="CENTER" hgap="5.0">
                    <columnConstraints>
                      <ColumnConstraints hgrow="SOMETIMES" maxWidth="150.0" minWidth="10.0" prefWidth="105.0" />
                      <ColumnConstraints hgrow="SOMETIMES" maxWidth="100.0" minWidth="10.0" prefWidth="85.0" />
                        <ColumnConstraints hgrow="SOMETIMES" maxWidth="100.0" minWidth="10.0" prefWidth="90.0" />
                        <ColumnConstraints hgrow="SOMETIMES" maxWidth="100.0" minWidth="10.0" prefWidth="85.0" />
                    </columnConstraints>
                    <rowConstraints>
                      <RowConstraints minHeight="10.0" prefHeight="30.0" vgrow="SOMETIMES" />
                    </rowConstraints>
                     <children>
                        <Label prefHeight="17.0" prefWidth="83.0" text="Interest Earned:" />
                        <Label text="Future Value:" GridPane.columnIndex="2" />
                        <TextField fx:id="txtInterestEarned" alignment="CENTER_RIGHT" editable="false" text="\$0.00"
```

```
GridPane.columnIndex="1" />
                        <TextField fx:id="txtFutureValue" alignment="CENTER_RIGHT" editable="false" text="\$0.00" GridPane.columnIndex="3" />
                     </children>
                     <HBox.margin>
                        <Insets bottom="10.0" left="20.0" right="20.0" top="10.0" />
                     </HBox.margin>
                  </GridPane>
               </children>
            </HBox>
            <Label alignment="CENTER" prefHeight="17.0" prefWidth="50.0" style="-fx-background-color: #fff;" text="Results" translateX="-145.0" translateY="-27.0" />
         </children>
      </StackPane>
      <HBox alignment="BOTTOM_RIGHT" maxWidth="345.0" prefHeight="21.0" prefWidth="410.0">
         <children>
            <Button fx:id="btnClose" alignment="CENTER" contentDisplay="CENTER" mnemonicParsing="false" onAction="#close" prefHeight="25.0" prefWidth="108.0" text="Close" />
         </children>
      </HBox>
   </children>
</VBox>
```

```java
package com.example.vphw04;

import javafx.application.Platform;
import javafx.beans.InvalidationListener;
import javafx.beans.property.*;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.fxml.FXML;
import javafx.scene.control.Button;
import javafx.scene.control.DatePicker;
import javafx.scene.control.TextField;

import java.math.RoundingMode;
import java.text.DecimalFormat;
import java.time.Instant;
import java.time.LocalDate;
import java.time.ZoneId;
import java.util.Date;
import java.util.concurrent.TimeUnit;

public class HelloController {

    @FXML
    private Button btnClose;

    @FXML
    private DatePicker dtpLoanEndDate;

    @FXML
    private DatePicker dtpStartDate;

    @FXML
    private TextField txtFutureValue;

    @FXML
    private TextField txtInterestEarned;

    @FXML
    private TextField txtInterestRate;

    @FXML
    private TextField txtPeriods;

    @FXML
    private TextField txtPrincipal;

    @FXML
    void initialize() {
        txtPrincipal.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(final ObservableValue<? extends String> observable, final String oldValue, final String newValue) {
                calculate();
            }
        });
        txtInterestRate.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(final ObservableValue<? extends String> observable, final String oldValue, final String newValue) {
                calculate();
            }
        });
        txtPeriods.textProperty().addListener(new ChangeListener<String>() {
            @Override
            public void changed(final ObservableValue<? extends String> observable, final String oldValue, final String newValue) {
                calculate();
            }
        });

        dtpStartDate.setValue(LocalDate.now());
        dtpLoanEndDate.setValue(LocalDate.now());

        dtpStartDate.valueProperty().addListener(new ChangeListener<LocalDate>() {
            @Override
            public void changed(ObservableValue<? extends LocalDate> observable, LocalDate oldValue, LocalDate newValue) {
                calculate();
            }
        });

        dtpLoanEndDate.valueProperty().addListener(new ChangeListener<LocalDate>() {
            @Override
            public void changed(ObservableValue<? extends LocalDate> observable, LocalDate oldValue, LocalDate newValue) {
                calculate();
            }
        });
    }

    void calculate(){
        double dPrincipal = 0.00D, dInterestRate = 0.00D,
                dPeriods = 0.00D;
        double dInterestEarned, dFutureValue;
        double periodsValue = 0;
        Date startDate, endDate;

        //Get the value of principal
        try {
            dPrincipal = Double.parseDouble(txtPrincipal.getText().trim());
            txtPrincipal.setStyle("-fx-border-color: #d3d3d3; -fx-border-radius: 3px;");
        } catch (NumberFormatException err) {
            txtPrincipal.setStyle("-fx-border-color: #f00; -fx-border-radius: 3px;");
            return;
        }

        // Get the interest rate
        try {
            dInterestRate = Double.parseDouble(txtInterestRate.getText().trim()) / 100;
            txtInterestRate.setStyle("-fx-border-color: #d3d3d3; -fx-border-radius: 3px;");
        } catch (NumberFormatException err) {
            txtInterestRate.setStyle("-fx-border-color: #f00; -fx-border-radius: 3px;");
            return;
        }

        DoubleProperty principal = new SimpleDoubleProperty(dPrincipal);
        DoubleProperty interestRate = new SimpleDoubleProperty(dInterestRate);

        // Get the start and end dates
        LocalDate localDateStart = dtpStartDate.getValue();
        Instant instantStart = Instant.from(localDateStart.atStartOfDay(ZoneId.systemDefault()));
        startDate = Date.from(instantStart);

        LocalDate localDateEnd = dtpLoanEndDate.getValue();
        Instant instantEnd = Instant.from(localDateEnd.atStartOfDay(ZoneId.systemDefault()));
        endDate = Date.from(instantEnd);

        // Make sure the end date doesn't occur before the start date
        if (startDate.after(endDate)){
            dtpLoanEndDate.setStyle("-fx-border-color: #f00; -fx-border-radius: 3px;");
            dtpLoanEndDate.setValue(dtpStartDate.getValue());
            return;
        }else{
            dtpLoanEndDate.setStyle("-fx-border-color: #d3d3d3; -fx-border-radius: 3px;");
        }

        // Get the difference in days
        // will be the periods
        long diffInMillies = Math.abs(endDate.getTime() - startDate.getTime());
        long days = TimeUnit.DAYS.convert(diffInMillies, TimeUnit.MILLISECONDS);
        if(days == 0) days = 1;
        txtPeriods.setText(Long.toString(days));

        // Because we will allow the user to directly specify
        // the number of days, let's get the period from its text box
```

```java
        try {
            periodsValue = Double.parseDouble(txtPeriods.getText().trim());
            txtPeriods.setStyle("-fx-border-color: #d3d3d3; -fx-border-radius: 3px;");
        } catch (NumberFormatException err) {
            txtPeriods.setStyle("-fx-border-color: #f00; -fx-border-radius: 3px;");
            return;
        }

        DoubleProperty pVal = new SimpleDoubleProperty(periodsValue);

        // 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()));
    }

    @FXML
    void close(ActionEvent event) {
        Platform.exit();
    }

    public static String truncate(double input) {
        DecimalFormat decimalFormat = new DecimalFormat("0.00");
        decimalFormat.setRoundingMode(RoundingMode.HALF_UP);
        String formatResult = decimalFormat.format(input);
        return formatResult.trim();
    }
}

```