PP2 CW-Report

Name : N.D.S Ranaweera

UOW Id : w1790193

IIT Id : 2019745

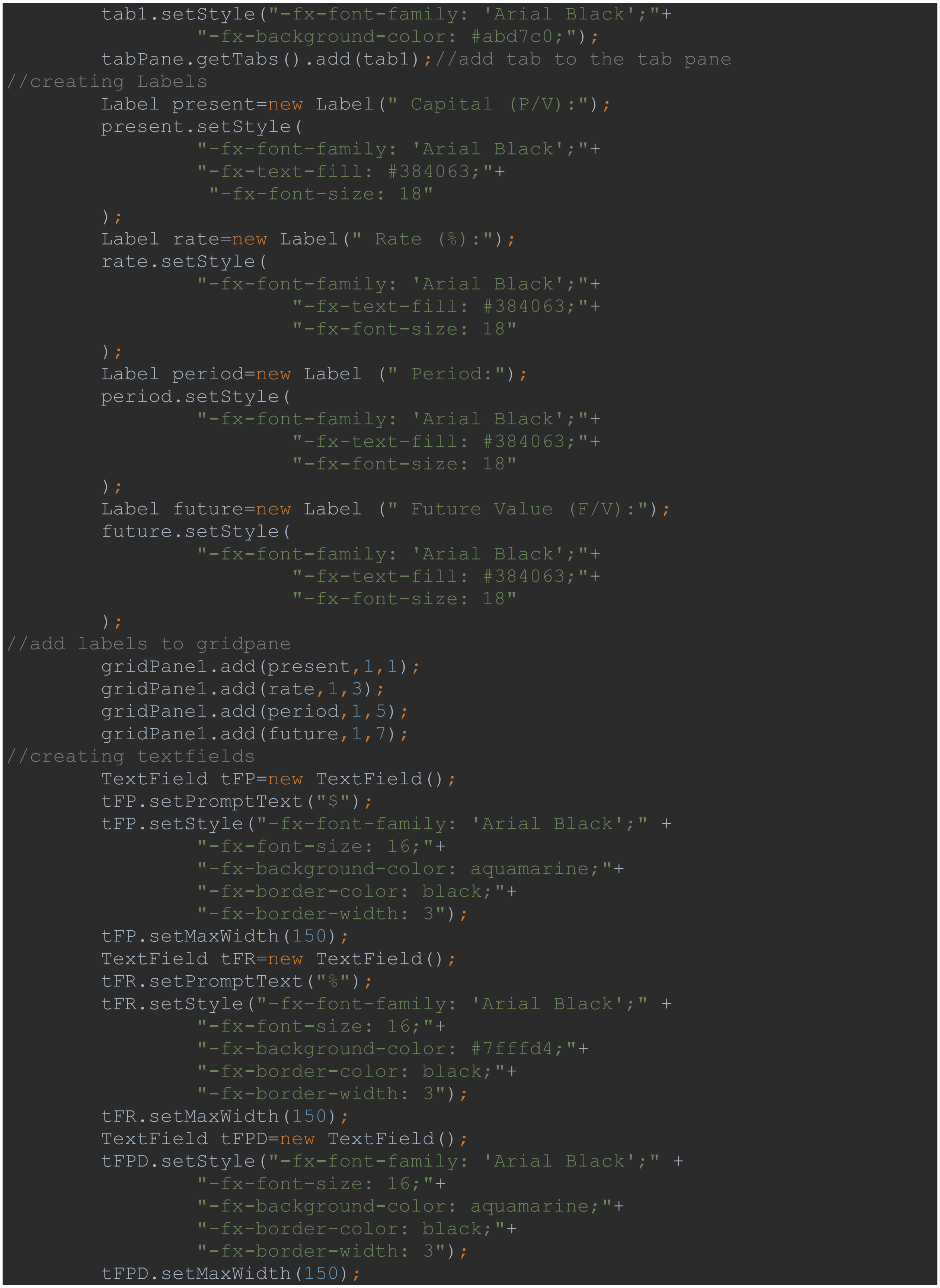
Introduction

Main class first create Tab pane and add 4 tabs and named them, then create 4 grid panes add them to each 4tabs one by one. Add All relevant labels and text fields and buttons to grid pane. Also added number pad to each grid pane.

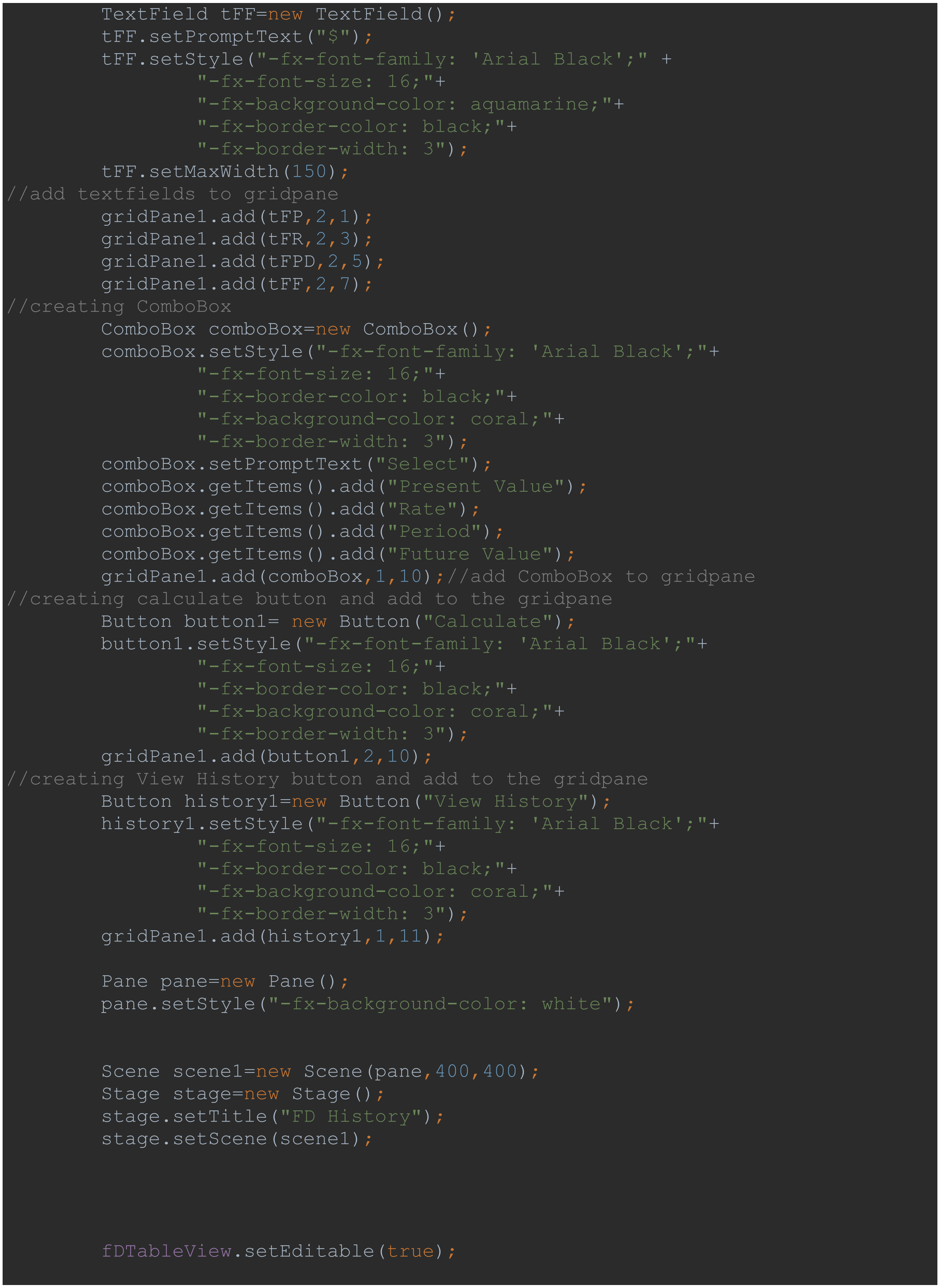
I used another class called "numberPad1" and inside the class create a method called "number Pad" and called method in Main class to set the number pad.

Main.java

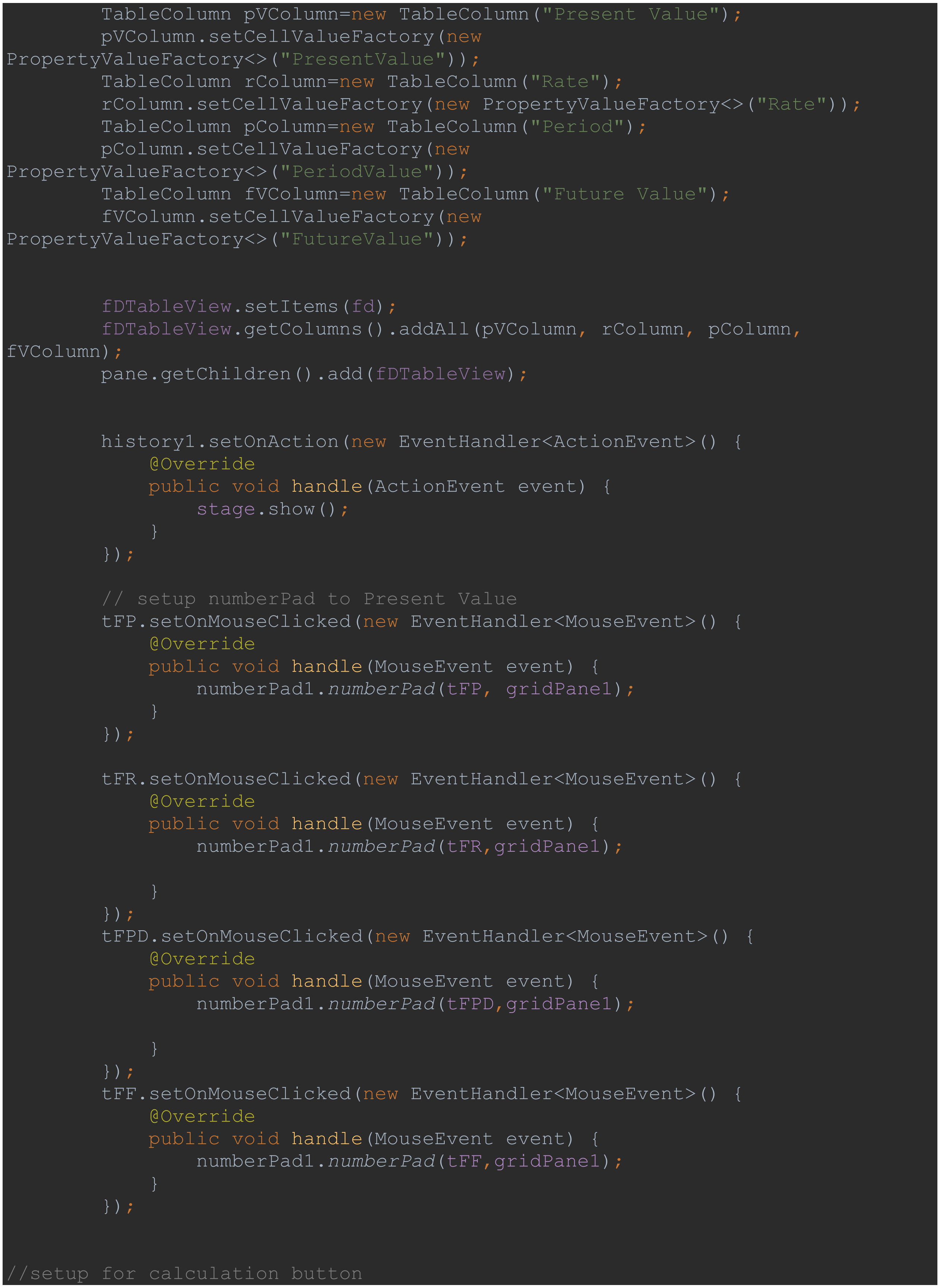
```
package sample;

import javafx.application.Application;
import javafx.beans.property.SimpleStringProperty;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.control.cell.PropertyValueFactory;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.GridPane;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;

import java.text.DecimalFormat;

public class Main extends Application {
    private final TableView<fixedD> fDTableView=new TableView<>();
    private final ObservableList<fixedD> fd=
FXCollections.observableArrayList();

    private final TableView<saving> savingTableView=new TableView<>();
    private final ObservableList<saving>
saving1=FXCollections.observableArrayList();

    private final TableView<loancal> loancalTableView=new TableView<>();
    private final ObservableList<loancal>
loan1=FXCollections.observableArrayList();

    private final TableView<mortgagecal> mortgagecalTableView=new
TableView<>();
    private final ObservableList<mortgagecal>
mortgage1=FXCollections.observableArrayList();

    @Override
    public void start(Stage primaryStage) throws Exception{


        DecimalFormat decimalFormat=new DecimalFormat("#0.00");//to get 2
decimal point
        primaryStage.setTitle("Financial Calculator");
//creating tabpane
        TabPane tabPane=new TabPane();
        tabPane.setStyle("-fx-border-color: black;"+
                "-fx-border-width: 3");
        GridPane gridPane1= new GridPane();//creat a GridPane
        gridPane1.setStyle("-fx-background-color: #b2bcff");
        gridPane1.setHgap(18);
        gridPane1.setVgap(18);
        Tab tab1=new Tab("Fixed Deposit",gridPane1);
```



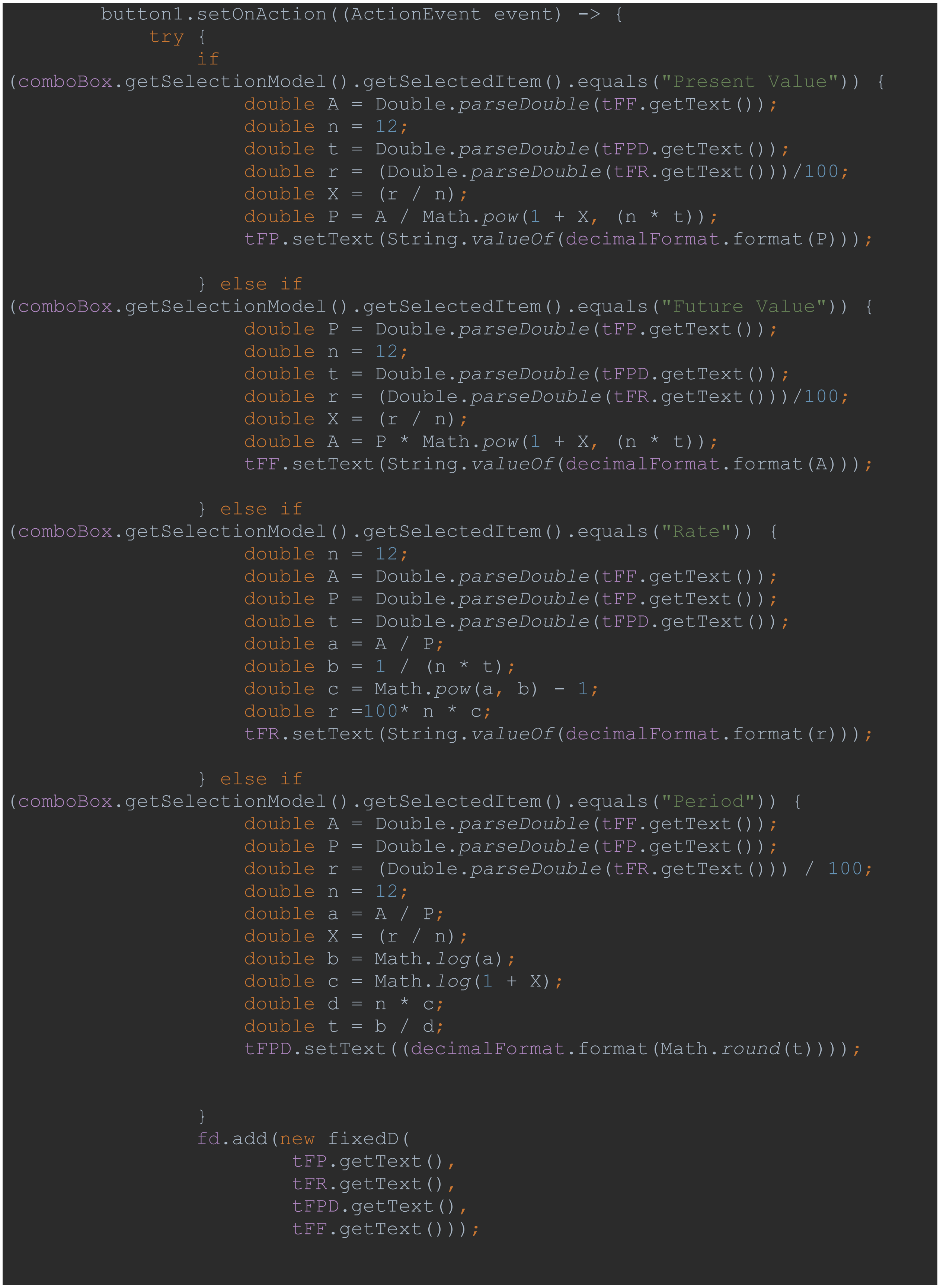
```
        tab1.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-background-color: #abd7c0;");
        tabPane.getTabs().add(tab1);//add tab to the tab pane
//creating Labels
        Label present=new Label(" Capital (P/V):");
        present.setStyle(
                "-fx-font-family: 'Arial Black';"+
                "-fx-text-fill: #384063;"+
                 "-fx-font-size: 18"
        );
        Label rate=new Label(" Rate (%):");
        rate.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: #384063;"+
                        "-fx-font-size: 18"
        );
        Label period=new Label (" Period:");
        period.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: #384063;"+
                        "-fx-font-size: 18"
        );
        Label future=new Label (" Future Value (F/V):");
        future.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: #384063;"+
                        "-fx-font-size: 18"
        );
//add labels to gridpane
        gridPane1.add(present,1,1);
        gridPane1.add(rate,1,3);
        gridPane1.add(period,1,5);
        gridPane1.add(future,1,7);
//creating textfields
        TextField tFP=new TextField();
        tFP.setPromptText("$");
        tFP.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aquamarine;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFP.setMaxWidth(150);
        TextField tFR=new TextField();
        tFR.setPromptText("%");
        tFR.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: #7fffd4;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFR.setMaxWidth(150);
        TextField tFPD=new TextField();
        tFPD.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aquamarine;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFPD.setMaxWidth(150);
```



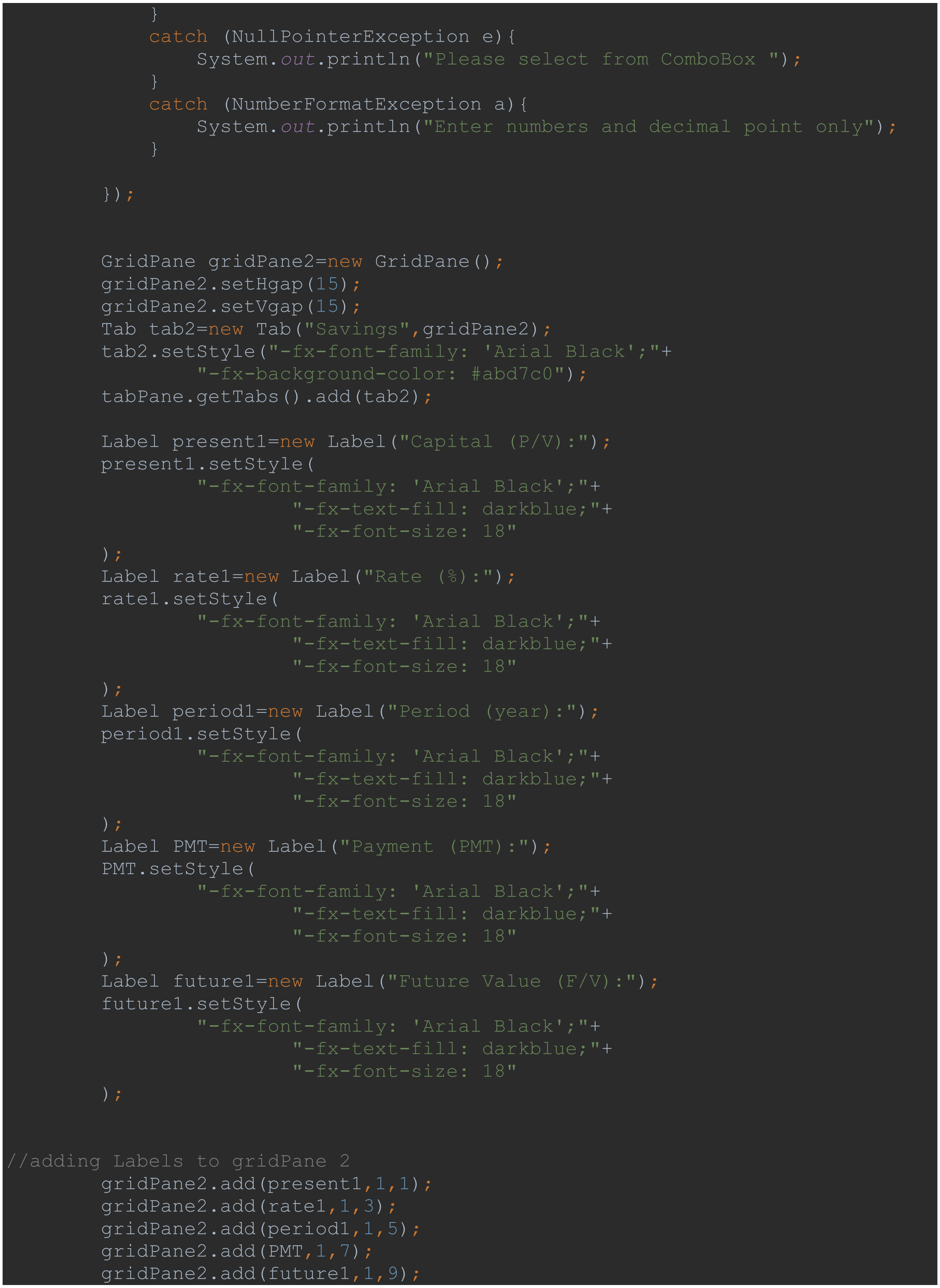
```
        TextField tFF=new TextField();
        tFF.setPromptText("$");
        tFF.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aquamarine;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFF.setMaxWidth(150);
//add textfields to gridpane
        gridPane1.add(tFP,2,1);
        gridPane1.add(tFR,2,3);
        gridPane1.add(tFPD,2,5);
        gridPane1.add(tFF,2,7);
//creating ComboBox
        ComboBox comboBox=new ComboBox();
        comboBox.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
        comboBox.setPromptText("Select");
        comboBox.getItems().add("Present Value");
        comboBox.getItems().add("Rate");
        comboBox.getItems().add("Period");
        comboBox.getItems().add("Future Value");
        gridPane1.add(comboBox,1,10);//add ComboBox to gridpane
//creating calculate button and add to the gridpane
        Button button1= new Button("Calculate");
        button1.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
        gridPane1.add(button1,2,10);
//creating View History button and add to the gridpane
        Button history1=new Button("View History");
        history1.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
        gridPane1.add(history1,1,11);

        Pane pane=new Pane();
        pane.setStyle("-fx-background-color: white");


        Scene scene1=new Scene(pane,400,400);
        Stage stage=new Stage();
        stage.setTitle("FD History");
        stage.setScene(scene1);




        fDTableView.setEditable(true);
```



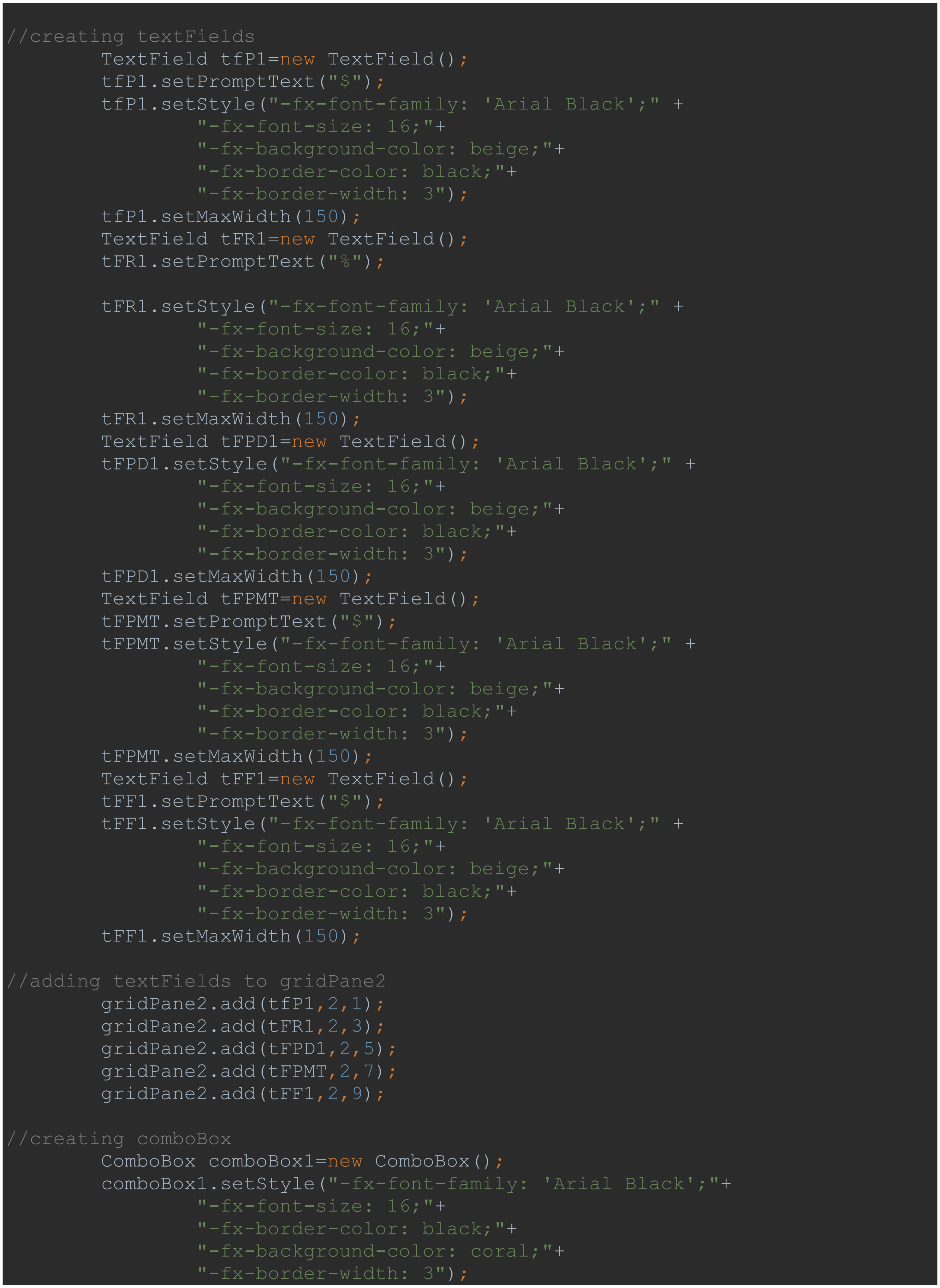
```
        TableColumn pVColumn=new TableColumn("Present Value");
        pVColumn.setCellValueFactory(new
PropertyValueFactory<>("PresentValue"));
        TableColumn rColumn=new TableColumn("Rate");
        rColumn.setCellValueFactory(new PropertyValueFactory<>("Rate"));
        TableColumn pColumn=new TableColumn("Period");
        pColumn.setCellValueFactory(new
PropertyValueFactory<>("PeriodValue"));
        TableColumn fVColumn=new TableColumn("Future Value");
        fVColumn.setCellValueFactory(new
PropertyValueFactory<>("FutureValue"));


        fDTableView.setItems(fd);
        fDTableView.getColumns().addAll(pVColumn, rColumn, pColumn,
fVColumn);
        pane.getChildren().add(fDTableView);


        history1.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                stage.show();
            }
        });

        // setup numberPad to Present Value
        tFP.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFP, gridPane1);
            }
        });

        tFR.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFR,gridPane1);

            }
        });
        tFPD.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFPD,gridPane1);

            }
        });
        tFF.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFF,gridPane1);
            }
        });


//setup for calculation button
```



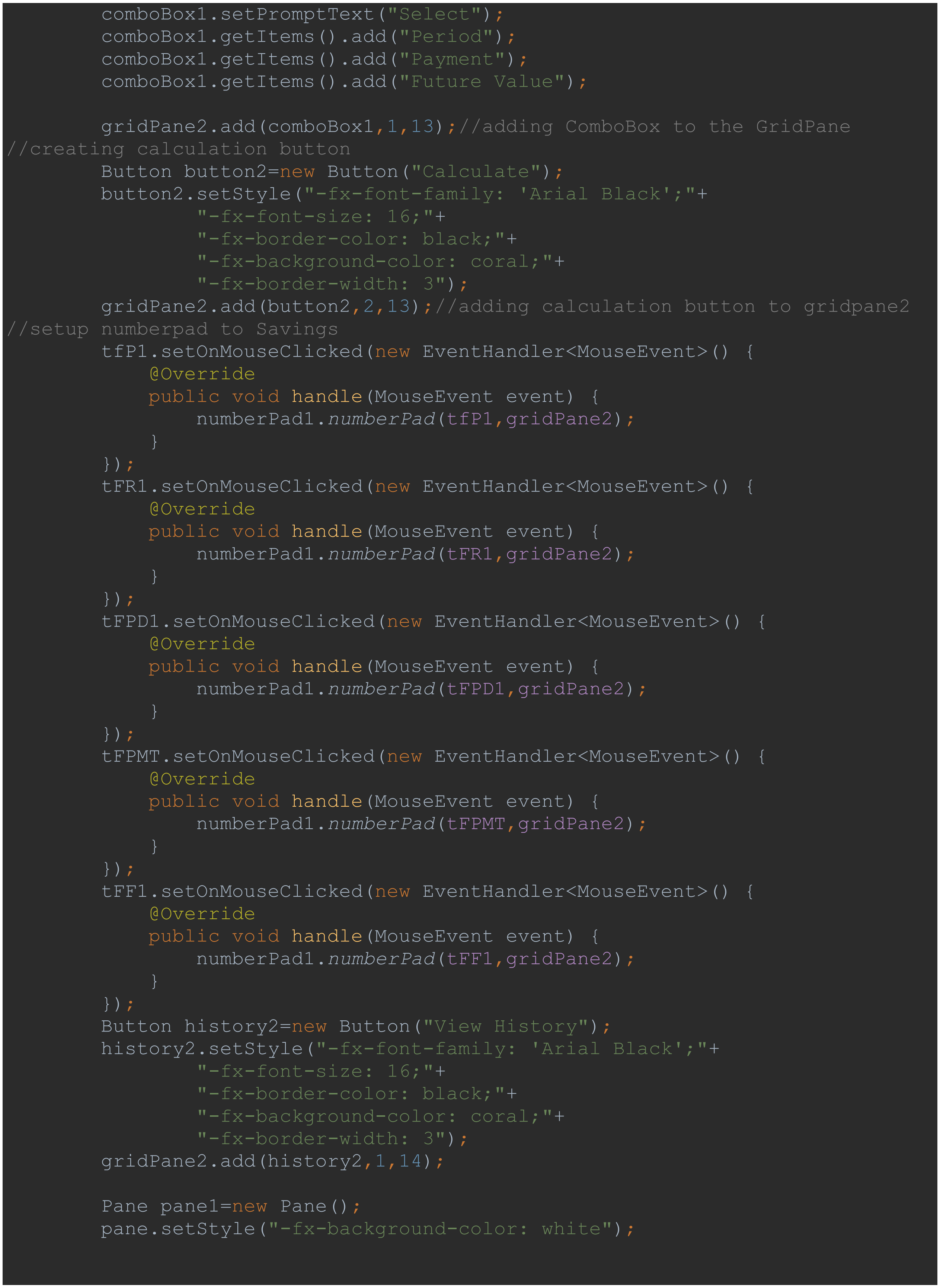
```
        button1.setOnAction((ActionEvent event) -> {
            try {
                if
(comboBox.getSelectionModel().getSelectedItem().equals("Present Value")) {
                    double A = Double.parseDouble(tFF.getText());
                    double n = 12;
                    double t = Double.parseDouble(tFPD.getText());
                    double r = (Double.parseDouble(tFR.getText()))/100;
                    double X = (r / n);
                    double P = A / Math.pow(1 + X, (n * t));
                    tFP.setText(String.valueOf(decimalFormat.format(P)));

                } else if
(comboBox.getSelectionModel().getSelectedItem().equals("Future Value")) {
                    double P = Double.parseDouble(tFP.getText());
                    double n = 12;
                    double t = Double.parseDouble(tFPD.getText());
                    double r = (Double.parseDouble(tFR.getText()))/100;
                    double X = (r / n);
                    double A = P * Math.pow(1 + X, (n * t));
                    tFF.setText(String.valueOf(decimalFormat.format(A)));

                } else if
(comboBox.getSelectionModel().getSelectedItem().equals("Rate")) {
                    double n = 12;
                    double A = Double.parseDouble(tFF.getText());
                    double P = Double.parseDouble(tFP.getText());
                    double t = Double.parseDouble(tFPD.getText());
                    double a = A / P;
                    double b = 1 / (n * t);
                    double c = Math.pow(a, b) - 1;
                    double r =100* n * c;
                    tFR.setText(String.valueOf(decimalFormat.format(r)));

                } else if
(comboBox.getSelectionModel().getSelectedItem().equals("Period")) {
                    double A = Double.parseDouble(tFF.getText());
                    double P = Double.parseDouble(tFP.getText());
                    double r = (Double.parseDouble(tFR.getText())) / 100;
                    double n = 12;
                    double a = A / P;
                    double X = (r / n);
                    double b = Math.log(a);
                    double c = Math.log(1 + X);
                    double d = n * c;
                    double t = b / d;
                    tFPD.setText((decimalFormat.format(Math.round(t))));


                }
                fd.add(new fixedD(
                        tFP.getText(),
                        tFR.getText(),
                        tFPD.getText(),
                        tFF.getText()));
```



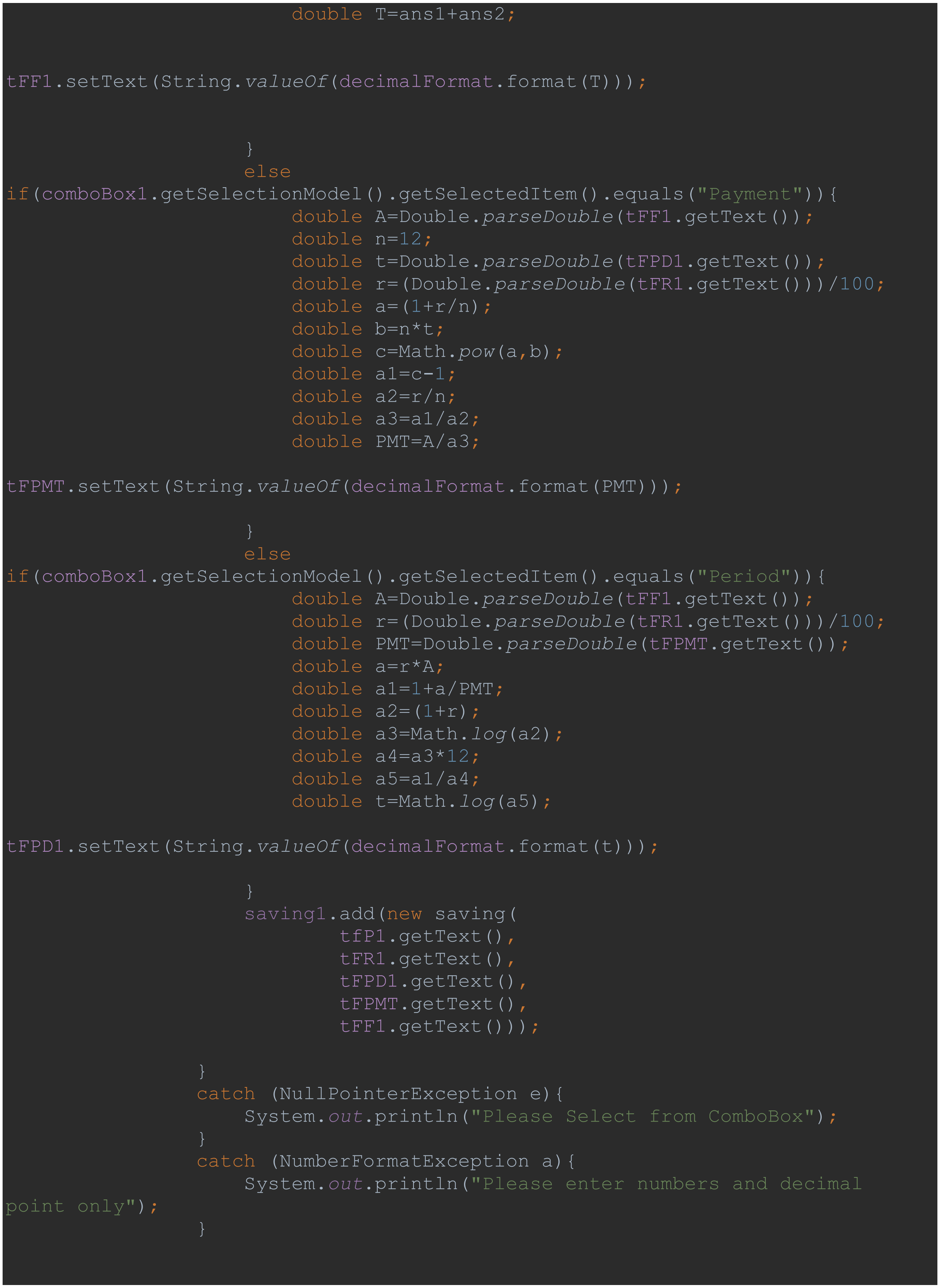
```
            }
            catch (NullPointerException e){
                System.out.println("Please select from ComboBox ");
            }
            catch (NumberFormatException a){
                System.out.println("Enter numbers and decimal point only");
            }

        });


        GridPane gridPane2=new GridPane();
        gridPane2.setHgap(15);
        gridPane2.setVgap(15);
        Tab tab2=new Tab("Savings",gridPane2);
        tab2.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-background-color: #abd7c0");
        tabPane.getTabs().add(tab2);

        Label present1=new Label("Capital (P/V):");
        present1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label rate1=new Label("Rate (%):");
        rate1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label period1=new Label("Period (year):");
        period1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label PMT=new Label("Payment (PMT):");
        PMT.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label future1=new Label("Future Value (F/V):");
        future1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );


//adding Labels to gridPane 2
        gridPane2.add(present1,1,1);
        gridPane2.add(rate1,1,3);
        gridPane2.add(period1,1,5);
        gridPane2.add(PMT,1,7);
        gridPane2.add(future1,1,9);
```



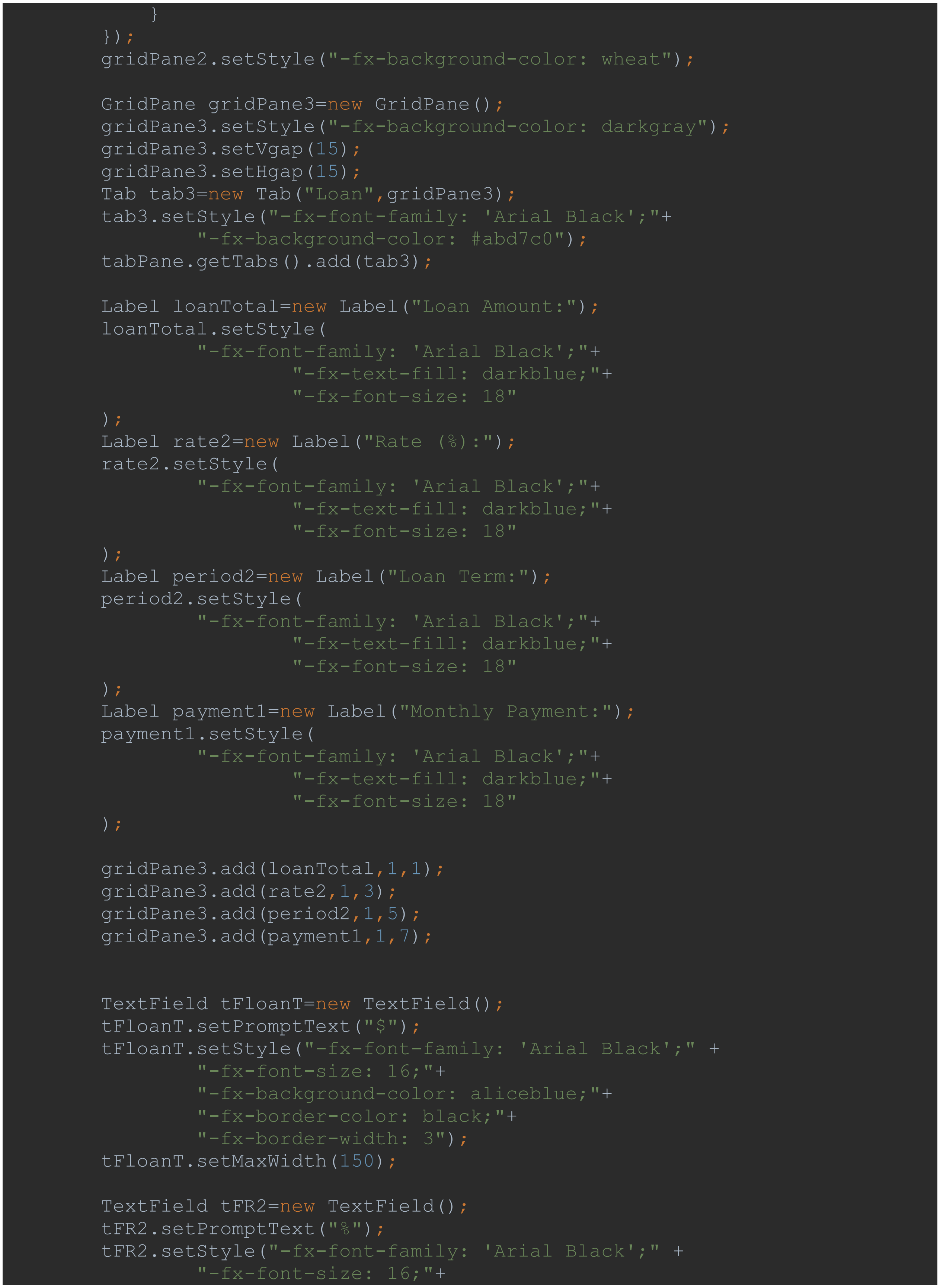
```
//creating textFields
        TextField tfP1=new TextField();
        tfP1.setPromptText("$");
        tfP1.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: beige;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tfP1.setMaxWidth(150);
        TextField tFR1=new TextField();
        tFR1.setPromptText("%");

        tFR1.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: beige;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFR1.setMaxWidth(150);
        TextField tFPD1=new TextField();
        tFPD1.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: beige;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFPD1.setMaxWidth(150);
        TextField tFPMT=new TextField();
        tFPMT.setPromptText("$");
        tFPMT.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: beige;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFPMT.setMaxWidth(150);
        TextField tFF1=new TextField();
        tFF1.setPromptText("$");
        tFF1.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: beige;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFF1.setMaxWidth(150);

//adding textFields to gridPane2
        gridPane2.add(tfP1,2,1);
        gridPane2.add(tFR1,2,3);
        gridPane2.add(tFPD1,2,5);
        gridPane2.add(tFPMT,2,7);
        gridPane2.add(tFF1,2,9);

//creating comboBox
        ComboBox comboBox1=new ComboBox();
        comboBox1.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
```



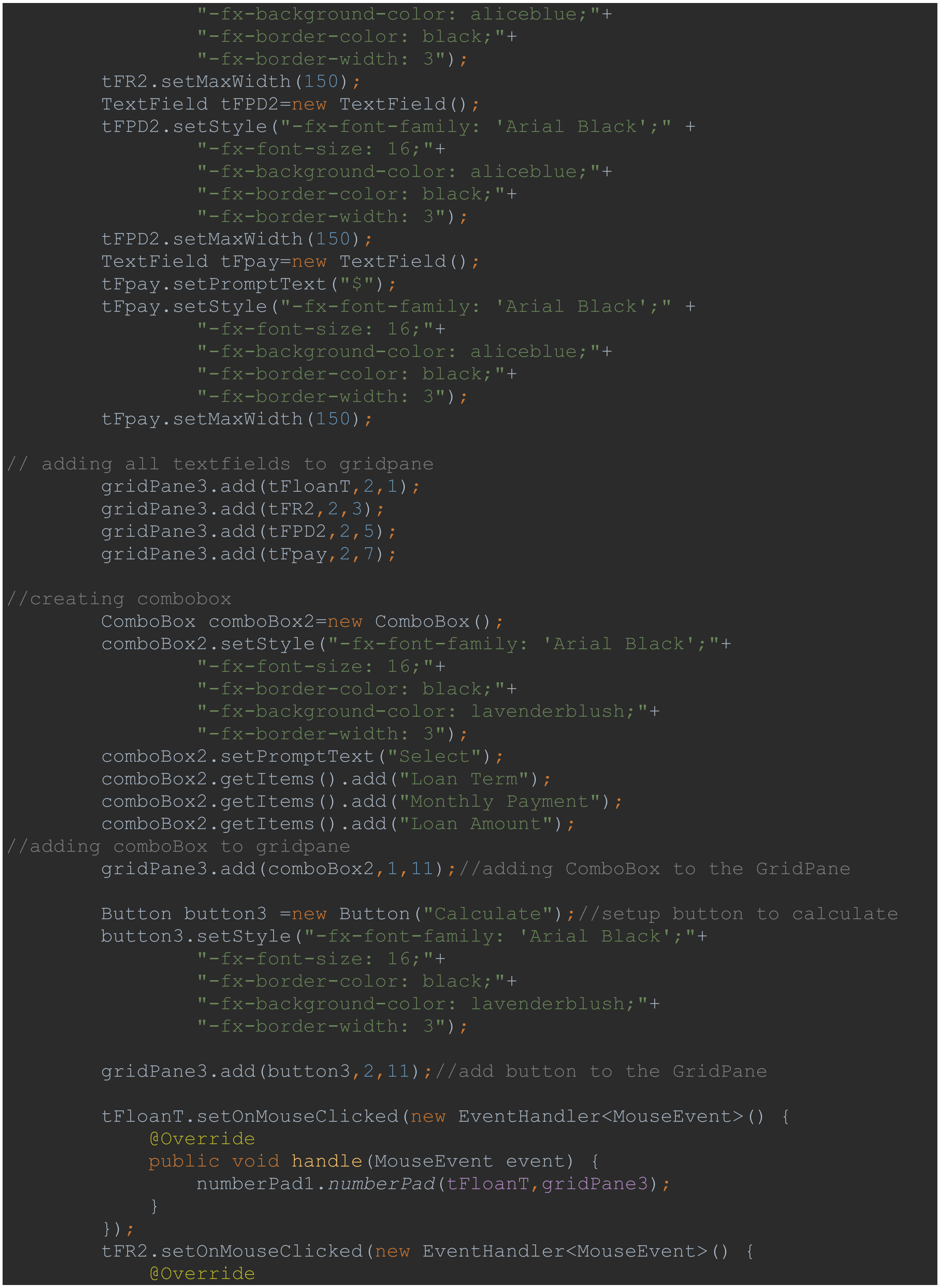
```
        comboBox1.setPromptText("Select");
        comboBox1.getItems().add("Period");
        comboBox1.getItems().add("Payment");
        comboBox1.getItems().add("Future Value");

        gridPane2.add(comboBox1,1,13);//adding ComboBox to the GridPane
//creating calculation button
        Button button2=new Button("Calculate");
        button2.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
        gridPane2.add(button2,2,13);//adding calculation button to gridpane2
//setup numberpad to Savings
        tfP1.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tfP1,gridPane2);
            }
        });
        tFR1.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFR1,gridPane2);
            }
        });
        tFPD1.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFPD1,gridPane2);
            }
        });
        tFPMT.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFPMT,gridPane2);
            }
        });
        tFF1.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFF1,gridPane2);
            }
        });
        Button history2=new Button("View History");
        history2.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: coral;"+
                "-fx-border-width: 3");
        gridPane2.add(history2,1,14);

        Pane pane1=new Pane();
        pane.setStyle("-fx-background-color: white");
        Scene scene2=new Scene(pane1,400,400);
        Stage stage2=new Stage();
        stage2.setTitle("Savings History");
        stage2.setScene(scene2);

        fDTableView.setEditable(true);

        TableColumn pVColumn1=new TableColumn("Present Value");
        pVColumn1.setCellValueFactory(new
PropertyValueFactory<>("PresentValue"));
        TableColumn rColumn1=new TableColumn("Rate");
        rColumn1.setCellValueFactory(new PropertyValueFactory<>("Rate"));
        TableColumn pColumn1=new TableColumn("Period");
        pColumn1.setCellValueFactory(new
PropertyValueFactory<>("PeriodValue"));
        TableColumn pmtColumn1=new TableColumn("Payment");
        pmtColumn1.setCellValueFactory(new
PropertyValueFactory<>("PaymentValue"));
        TableColumn fVColumn1=new TableColumn("Future Value");
        fVColumn1.setCellValueFactory(new
PropertyValueFactory<>("FutureValue"));


        savingTableView.setItems(saving1);
        savingTableView.getColumns().addAll(pVColumn1, rColumn1,
pColumn1,pmtColumn1, fVColumn1);
        pane1.getChildren().add(savingTableView);

        history2.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                stage2.show();
            }
        });



//set on action to calculation button to calculate values
        button2.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                try {

if(comboBox1.getSelectionModel().getSelectedItem().equals("Future Value")){
                        double P=Double.parseDouble(tfP1.getText());
                        double r=(Double.parseDouble(tFR1.getText()))/100;
                        double PMT=Double.parseDouble(tFPMT.getText());
                        double n=12;
                        double t=Double.parseDouble(tFPD1.getText());
                        double a=(1+r/n);
                        double b=n*t;
                        double c=Math.pow(a,b);
                        double ans1=P*c;
                        double a1=c-1;
                        double a2=r/n;
                        double a3=a1/a2;
                        double ans2=PMT*a3;
```



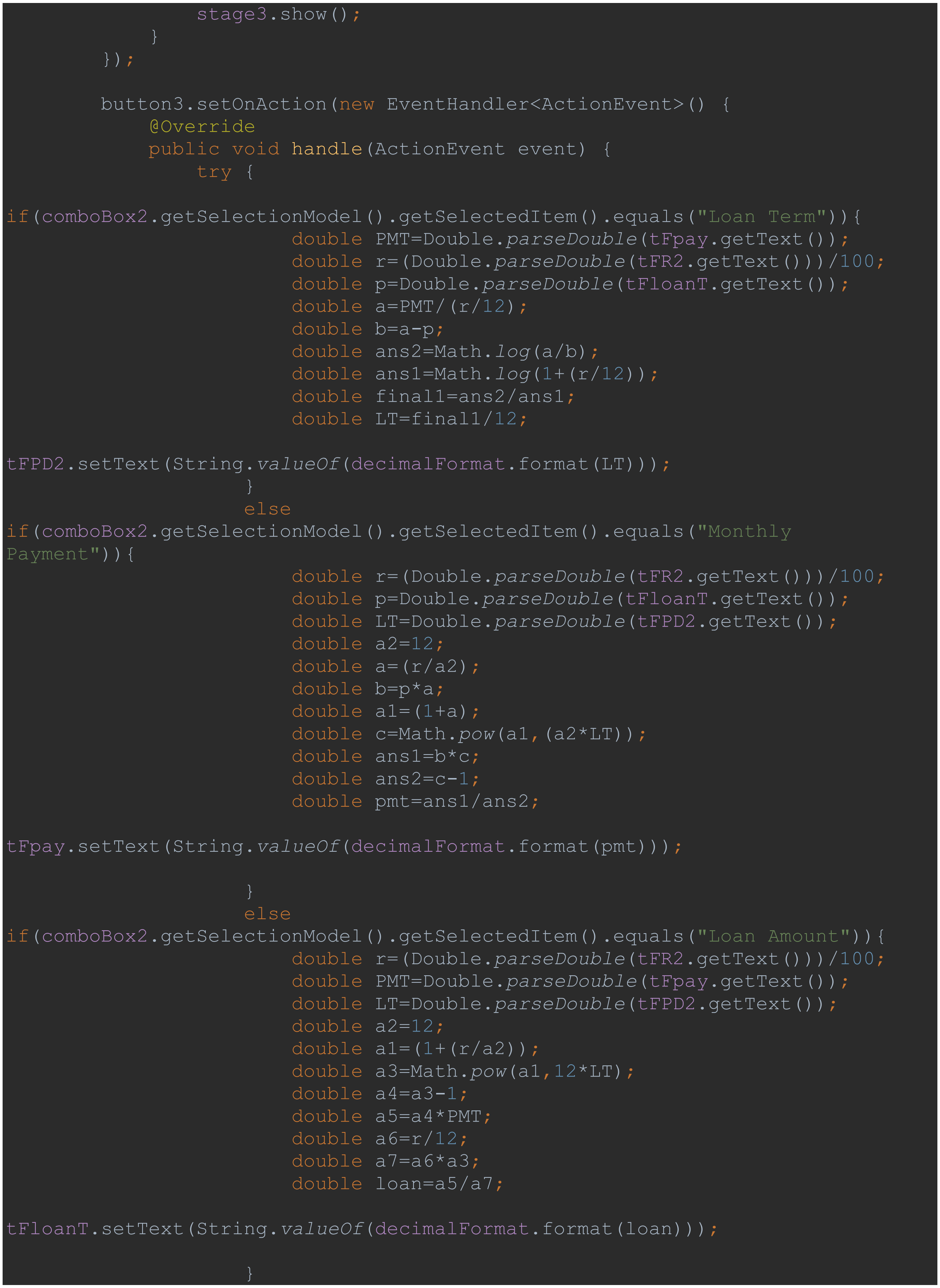
```
                        double T=ans1+ans2;


tFF1.setText(String.valueOf(decimalFormat.format(T)));


                    }
                    else
if(comboBox1.getSelectionModel().getSelectedItem().equals("Payment")){
                        double A=Double.parseDouble(tFF1.getText());
                        double n=12;
                        double t=Double.parseDouble(tFPD1.getText());
                        double r=(Double.parseDouble(tFR1.getText()))/100;
                        double a=(1+r/n);
                        double b=n*t;
                        double c=Math.pow(a,b);
                        double a1=c-1;
                        double a2=r/n;
                        double a3=a1/a2;
                        double PMT=A/a3;

tFPMT.setText(String.valueOf(decimalFormat.format(PMT)));

                    }
                    else
if(comboBox1.getSelectionModel().getSelectedItem().equals("Period")){
                        double A=Double.parseDouble(tFF1.getText());
                        double r=(Double.parseDouble(tFR1.getText()))/100;
                        double PMT=Double.parseDouble(tFPMT.getText());
                        double a=r*A;
                        double a1=1+a/PMT;
                        double a2=(1+r);
                        double a3=Math.log(a2);
                        double a4=a3*12;
                        double a5=a1/a4;
                        double t=Math.log(a5);

tFPD1.setText(String.valueOf(decimalFormat.format(t)));

                    }
                    saving1.add(new saving(
                            tfP1.getText(),
                            tFR1.getText(),
                            tFPD1.getText(),
                            tFPMT.getText(),
                            tFF1.getText()));

                }
                catch (NullPointerException e){
                    System.out.println("Please Select from ComboBox");
                }
                catch (NumberFormatException a){
                    System.out.println("Please enter numbers and decimal
point only");
                }
```



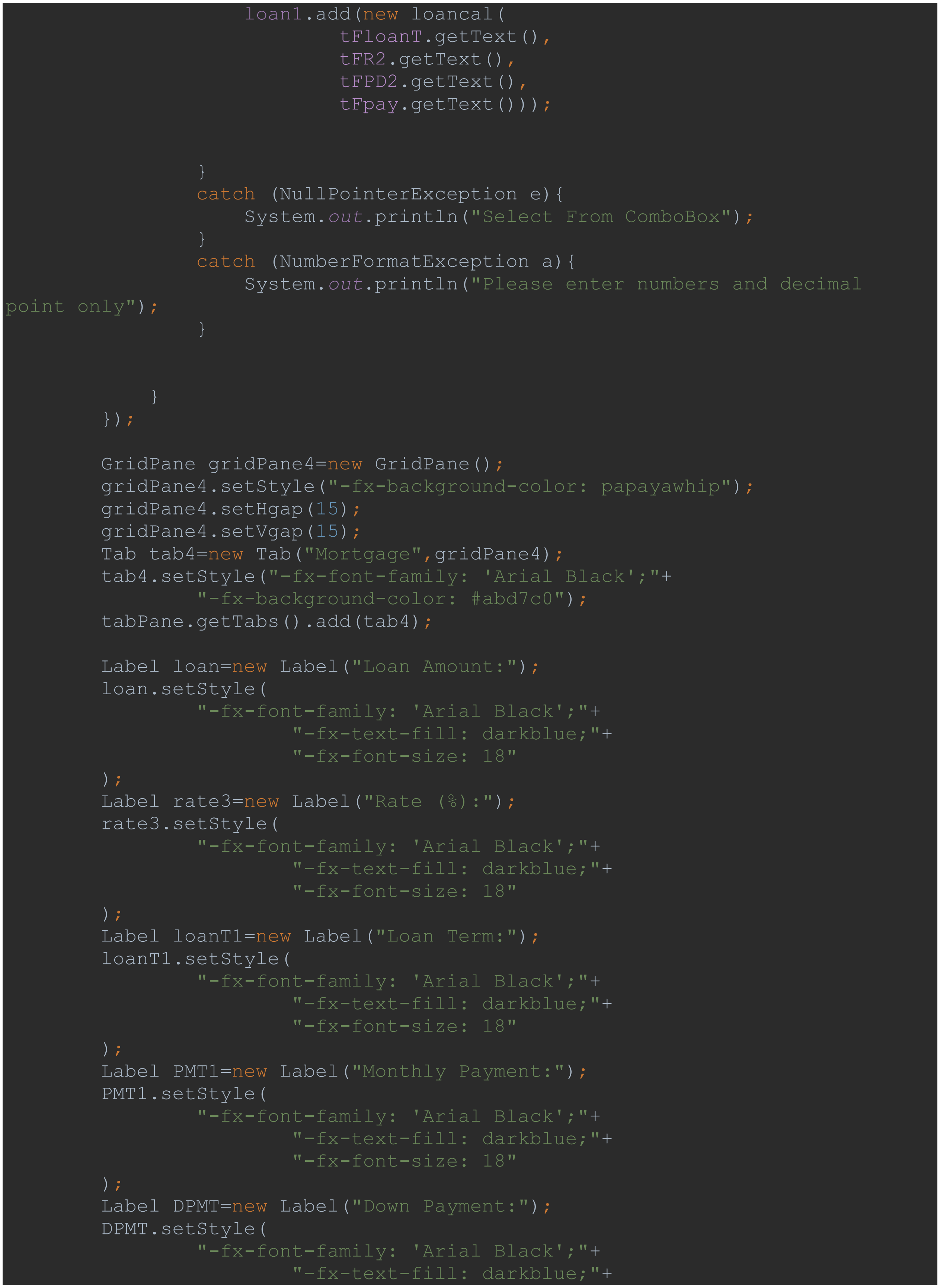
```
            }
        });
        gridPane2.setStyle("-fx-background-color: wheat");

        GridPane gridPane3=new GridPane();
        gridPane3.setStyle("-fx-background-color: darkgray");
        gridPane3.setVgap(15);
        gridPane3.setHgap(15);
        Tab tab3=new Tab("Loan",gridPane3);
        tab3.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-background-color: #abd7c0");
        tabPane.getTabs().add(tab3);

        Label loanTotal=new Label("Loan Amount:");
        loanTotal.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label rate2=new Label("Rate (%):");
        rate2.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label period2=new Label("Loan Term:");
        period2.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label payment1=new Label("Monthly Payment:");
        payment1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );

        gridPane3.add(loanTotal,1,1);
        gridPane3.add(rate2,1,3);
        gridPane3.add(period2,1,5);
        gridPane3.add(payment1,1,7);


        TextField tFloanT=new TextField();
        tFloanT.setPromptText("$");
        tFloanT.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFloanT.setMaxWidth(150);

        TextField tFR2=new TextField();
        tFR2.setPromptText("%");
        tFR2.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
```



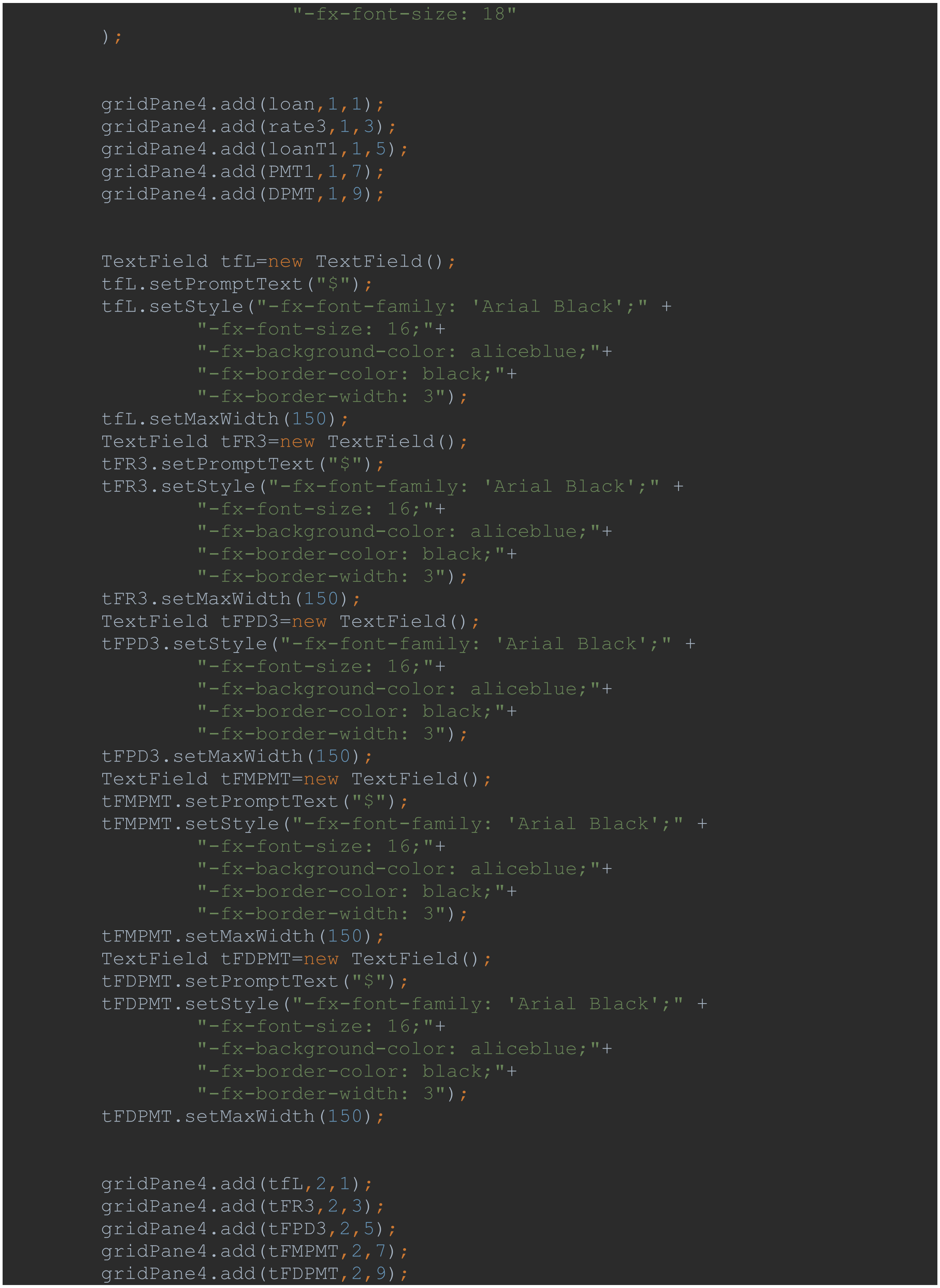
```
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFR2.setMaxWidth(150);
        TextField tFPD2=new TextField();
        tFPD2.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFPD2.setMaxWidth(150);
        TextField tFpay=new TextField();
        tFpay.setPromptText("$");
        tFpay.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFpay.setMaxWidth(150);

// adding all textfields to gridpane
        gridPane3.add(tFloanT,2,1);
        gridPane3.add(tFR2,2,3);
        gridPane3.add(tFPD2,2,5);
        gridPane3.add(tFpay,2,7);

//creating combobox
        ComboBox comboBox2=new ComboBox();
        comboBox2.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: lavenderblush;"+
                "-fx-border-width: 3");
        comboBox2.setPromptText("Select");
        comboBox2.getItems().add("Loan Term");
        comboBox2.getItems().add("Monthly Payment");
        comboBox2.getItems().add("Loan Amount");
//adding comboBox to gridpane
        gridPane3.add(comboBox2,1,11);//adding ComboBox to the GridPane

        Button button3 =new Button("Calculate");//setup button to calculate
        button3.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: lavenderblush;"+
                "-fx-border-width: 3");

        gridPane3.add(button3,2,11);//add button to the GridPane

        tFloanT.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFloanT,gridPane3);
            }
        });
        tFR2.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFR2,gridPane3);
            }
        });
        tFPD2.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFPD2,gridPane3);
            }
        });
        tFpay.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                numberPad1.numberPad(tFPMT,gridPane3);
            }
        });

        Button history3=new Button("View History");
        history3.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: lavenderblush;"+
                "-fx-border-width: 3");
        gridPane3.add(history3,1,12);

        Pane pane2=new Pane();
        pane2.setStyle("-fx-background-color: white");


        Scene scene3=new Scene(pane2,400,400);
        Stage stage3=new Stage();
        stage3.setTitle("Loan History");
        stage3.setScene(scene3);

        loancalTableView.setEditable(true);

        TableColumn lAColumn=new TableColumn("Loan Amount");
        lAColumn.setCellValueFactory(new
PropertyValueFactory<>("LoanAmount"));
        TableColumn rColumn2=new TableColumn("Rate");
        rColumn2.setCellValueFactory(new PropertyValueFactory<>("Rate"));
        TableColumn pColumn2=new TableColumn("Loan Term");
        pColumn2.setCellValueFactory(new PropertyValueFactory<>("LoanTerm"));
        TableColumn pmtColumn2=new TableColumn("MonthlyPayment");
        pmtColumn2.setCellValueFactory(new
PropertyValueFactory<>("MonthlyPayment"));



        loancalTableView.setItems(loan1);
        loancalTableView.getColumns().addAll(lAColumn, rColumn2, pColumn2,
pmtColumn2);
        pane2.getChildren().add(loancalTableView);

        history3.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
```



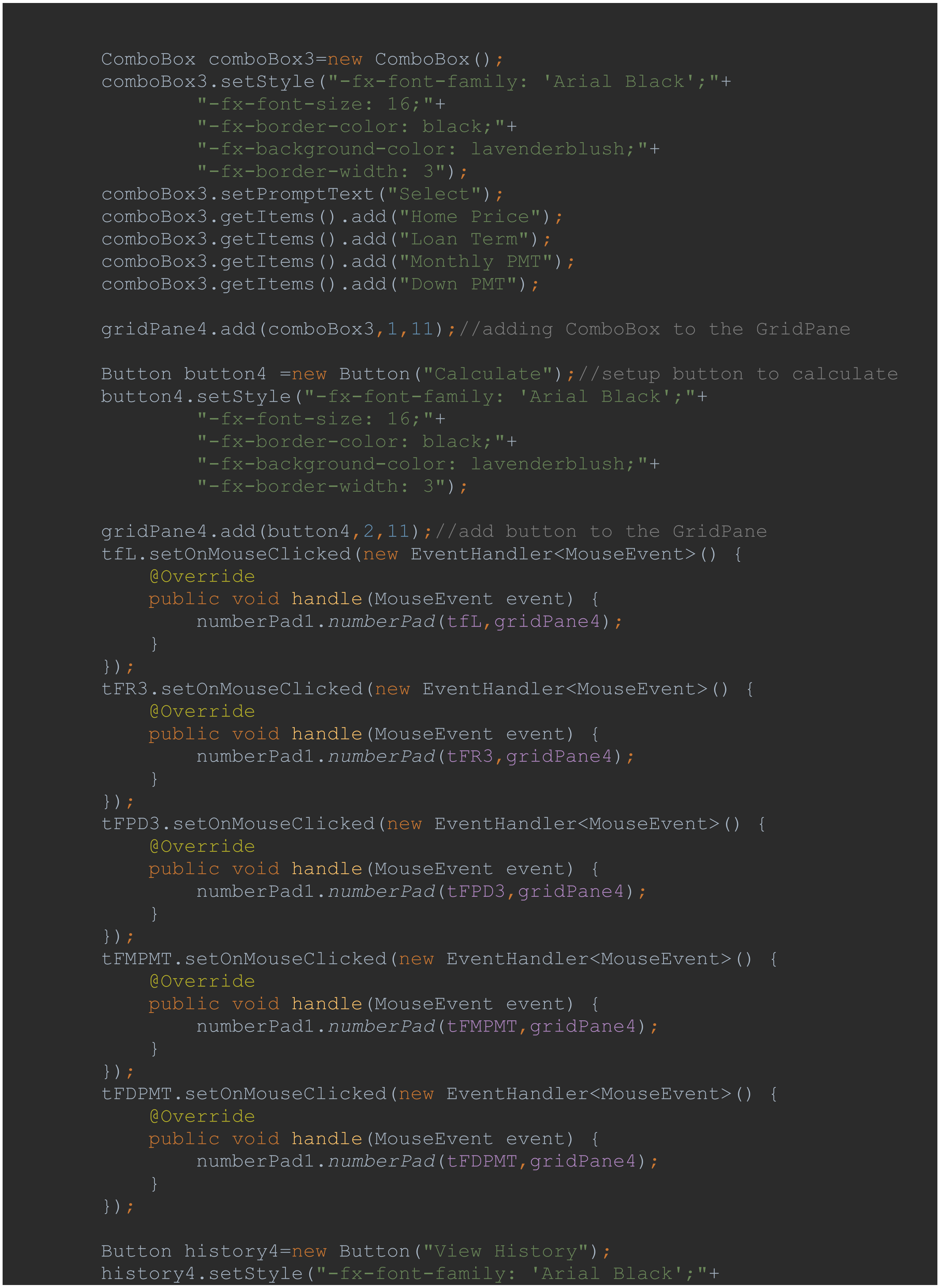
```
                stage3.show();
            }
        });

        button3.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                try {

if(comboBox2.getSelectionModel().getSelectedItem().equals("Loan Term")){
                        double PMT=Double.parseDouble(tFpay.getText());
                        double r=(Double.parseDouble(tFR2.getText()))/100;
                        double p=Double.parseDouble(tFloanT.getText());
                        double a=PMT/(r/12);
                        double b=a-p;
                        double ans2=Math.log(a/b);
                        double ans1=Math.log(1+(r/12));
                        double final1=ans2/ans1;
                        double LT=final1/12;

tFPD2.setText(String.valueOf(decimalFormat.format(LT)));
                    }
                    else
if(comboBox2.getSelectionModel().getSelectedItem().equals("Monthly
Payment")){
                        double r=(Double.parseDouble(tFR2.getText()))/100;
                        double p=Double.parseDouble(tFloanT.getText());
                        double LT=Double.parseDouble(tFPD2.getText());
                        double a2=12;
                        double a=(r/a2);
                        double b=p*a;
                        double a1=(1+a);
                        double c=Math.pow(a1,(a2*LT));
                        double ans1=b*c;
                        double ans2=c-1;
                        double pmt=ans1/ans2;

tFpay.setText(String.valueOf(decimalFormat.format(pmt)));

                    }
                    else
if(comboBox2.getSelectionModel().getSelectedItem().equals("Loan Amount")){
                        double r=(Double.parseDouble(tFR2.getText()))/100;
                        double PMT=Double.parseDouble(tFpay.getText());
                        double LT=Double.parseDouble(tFPD2.getText());
                        double a2=12;
                        double a1=(1+(r/a2));
                        double a3=Math.pow(a1,12*LT);
                        double a4=a3-1;
                        double a5=a4*PMT;
                        double a6=r/12;
                        double a7=a6*a3;
                        double loan=a5/a7;

tFloanT.setText(String.valueOf(decimalFormat.format(loan)));

                    }
```



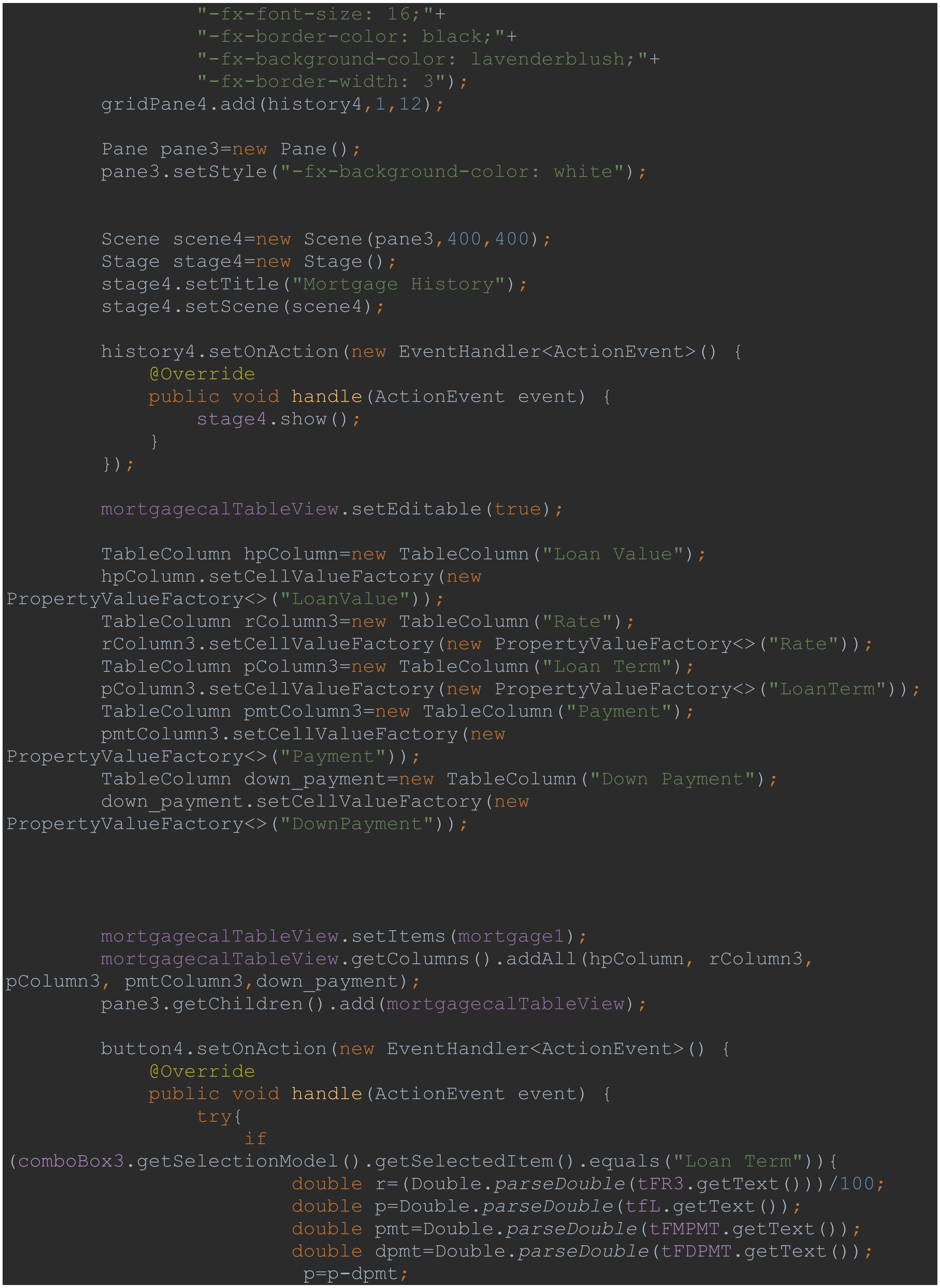
```
                    loan1.add(new loancal(
                            tFloanT.getText(),
                            tFR2.getText(),
                            tFPD2.getText(),
                            tFpay.getText()));


                }
                catch (NullPointerException e){
                    System.out.println("Select From ComboBox");
                }
                catch (NumberFormatException a){
                    System.out.println("Please enter numbers and decimal 
point only");
                }


            }
        });

        GridPane gridPane4=new GridPane();
        gridPane4.setStyle("-fx-background-color: papayawhip");
        gridPane4.setHgap(15);
        gridPane4.setVgap(15);
        Tab tab4=new Tab("Mortgage",gridPane4);
        tab4.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-background-color: #abd7c0");
        tabPane.getTabs().add(tab4);

        Label loan=new Label("Loan Amount:");
        loan.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label rate3=new Label("Rate (%):");
        rate3.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label loanT1=new Label("Loan Term:");
        loanT1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label PMT1=new Label("Monthly Payment:");
        PMT1.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
                        "-fx-font-size: 18"
        );
        Label DPMT=new Label("Down Payment:");
        DPMT.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: darkblue;"+
```



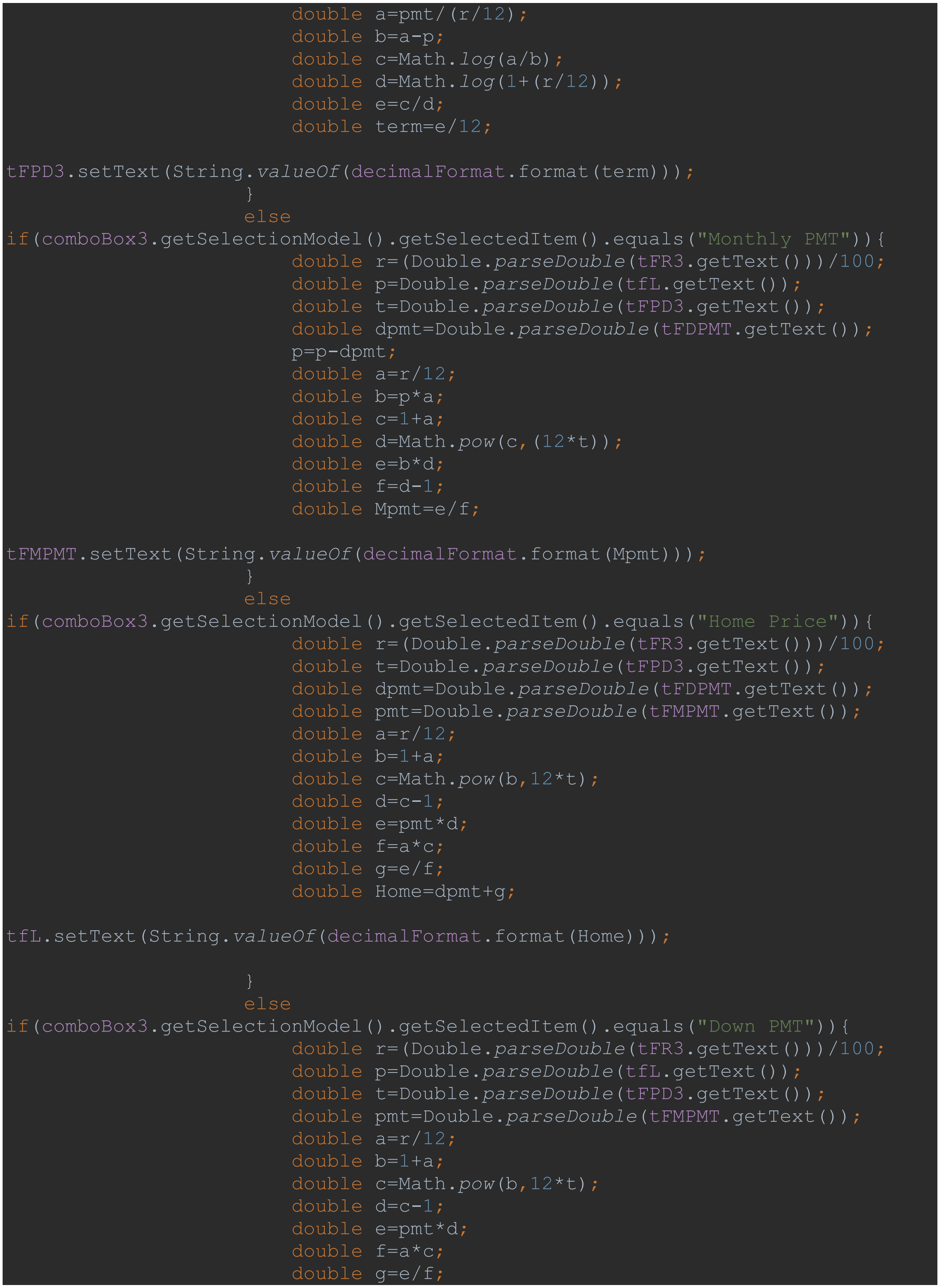
```
                        "-fx-font-size: 18"
        );


        gridPane4.add(loan,1,1);
        gridPane4.add(rate3,1,3);
        gridPane4.add(loanT1,1,5);
        gridPane4.add(PMT1,1,7);
        gridPane4.add(DPMT,1,9);


        TextField tfL=new TextField();
        tfL.setPromptText("$");
        tfL.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tfL.setMaxWidth(150);
        TextField tFR3=new TextField();
        tFR3.setPromptText("$");
        tFR3.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFR3.setMaxWidth(150);
        TextField tFPD3=new TextField();
        tFPD3.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFPD3.setMaxWidth(150);
        TextField tFMPMT=new TextField();
        tFMPMT.setPromptText("$");
        tFMPMT.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFMPMT.setMaxWidth(150);
        TextField tFDPMT=new TextField();
        tFDPMT.setPromptText("$");
        tFDPMT.setStyle("-fx-font-family: 'Arial Black';" +
                "-fx-font-size: 16;"+
                "-fx-background-color: aliceblue;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3");
        tFDPMT.setMaxWidth(150);


        gridPane4.add(tfL,2,1);
        gridPane4.add(tFR3,2,3);
        gridPane4.add(tFPD3,2,5);
        gridPane4.add(tFMPMT,2,7);
        gridPane4.add(tFDPMT,2,9);
```



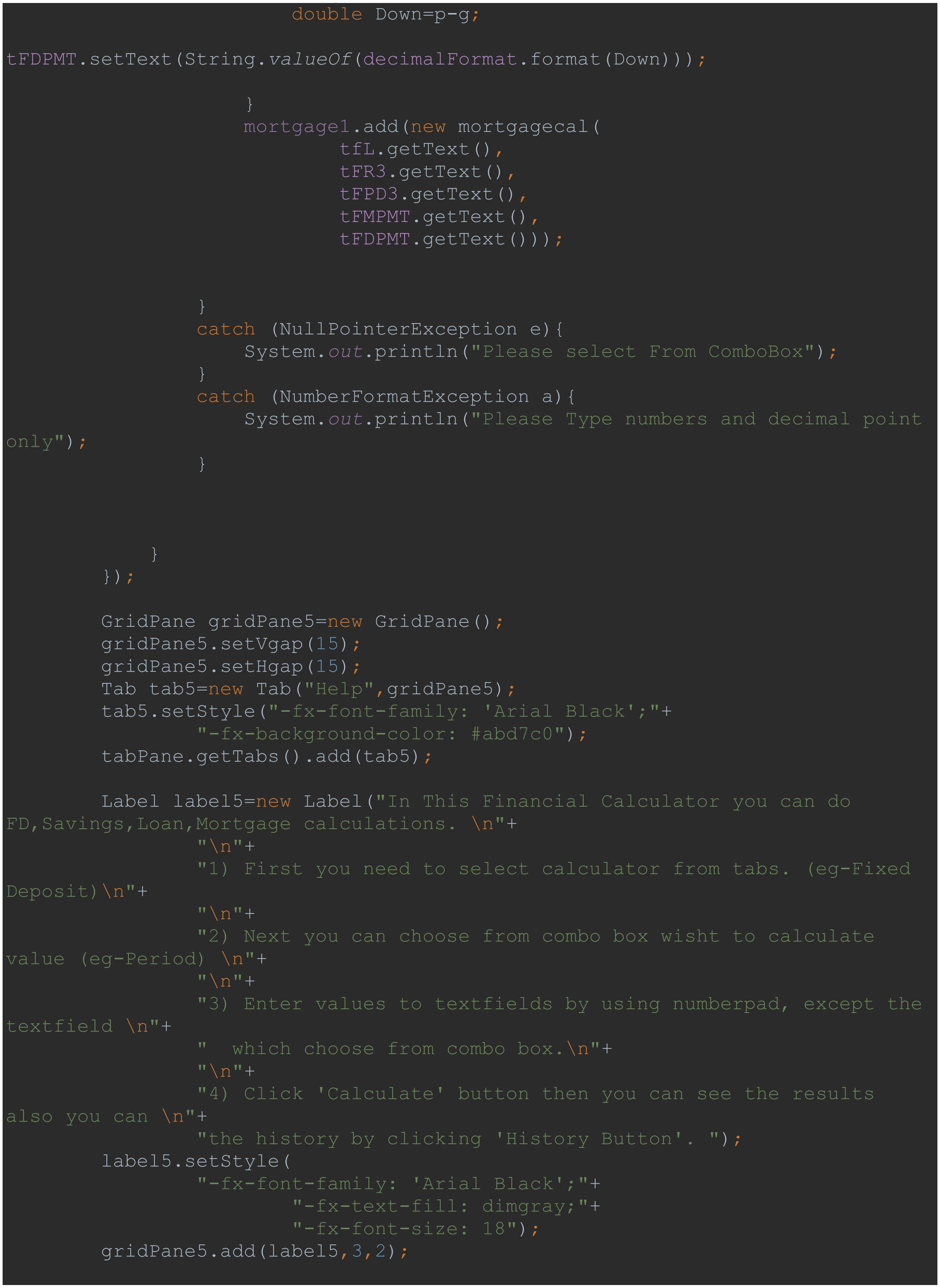
```
ComboBox comboBox3=new ComboBox();
comboBox3.setStyle("-fx-font-family: 'Arial Black';"+
        "-fx-font-size: 16;"+
        "-fx-border-color: black;"+
        "-fx-background-color: lavenderblush;"+
        "-fx-border-width: 3");
comboBox3.setPromptText("Select");
comboBox3.getItems().add("Home Price");
comboBox3.getItems().add("Loan Term");
comboBox3.getItems().add("Monthly PMT");
comboBox3.getItems().add("Down PMT");

gridPane4.add(comboBox3,1,11);//adding ComboBox to the GridPane

Button button4 =new Button("Calculate");//setup button to calculate
button4.setStyle("-fx-font-family: 'Arial Black';"+
        "-fx-font-size: 16;"+
        "-fx-border-color: black;"+
        "-fx-background-color: lavenderblush;"+
        "-fx-border-width: 3");

gridPane4.add(button4,2,11);//add button to the GridPane
tfL.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        numberPad1.numberPad(tfL,gridPane4);
    }
});
tFR3.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        numberPad1.numberPad(tFR3,gridPane4);
    }
});
tFPD3.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        numberPad1.numberPad(tFPD3,gridPane4);
    }
});
tFMPMT.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        numberPad1.numberPad(tFMPMT,gridPane4);
    }
});
tFDPMT.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        numberPad1.numberPad(tFDPMT,gridPane4);
    }
});

Button history4=new Button("View History");
history4.setStyle("-fx-font-family: 'Arial Black';"+
```



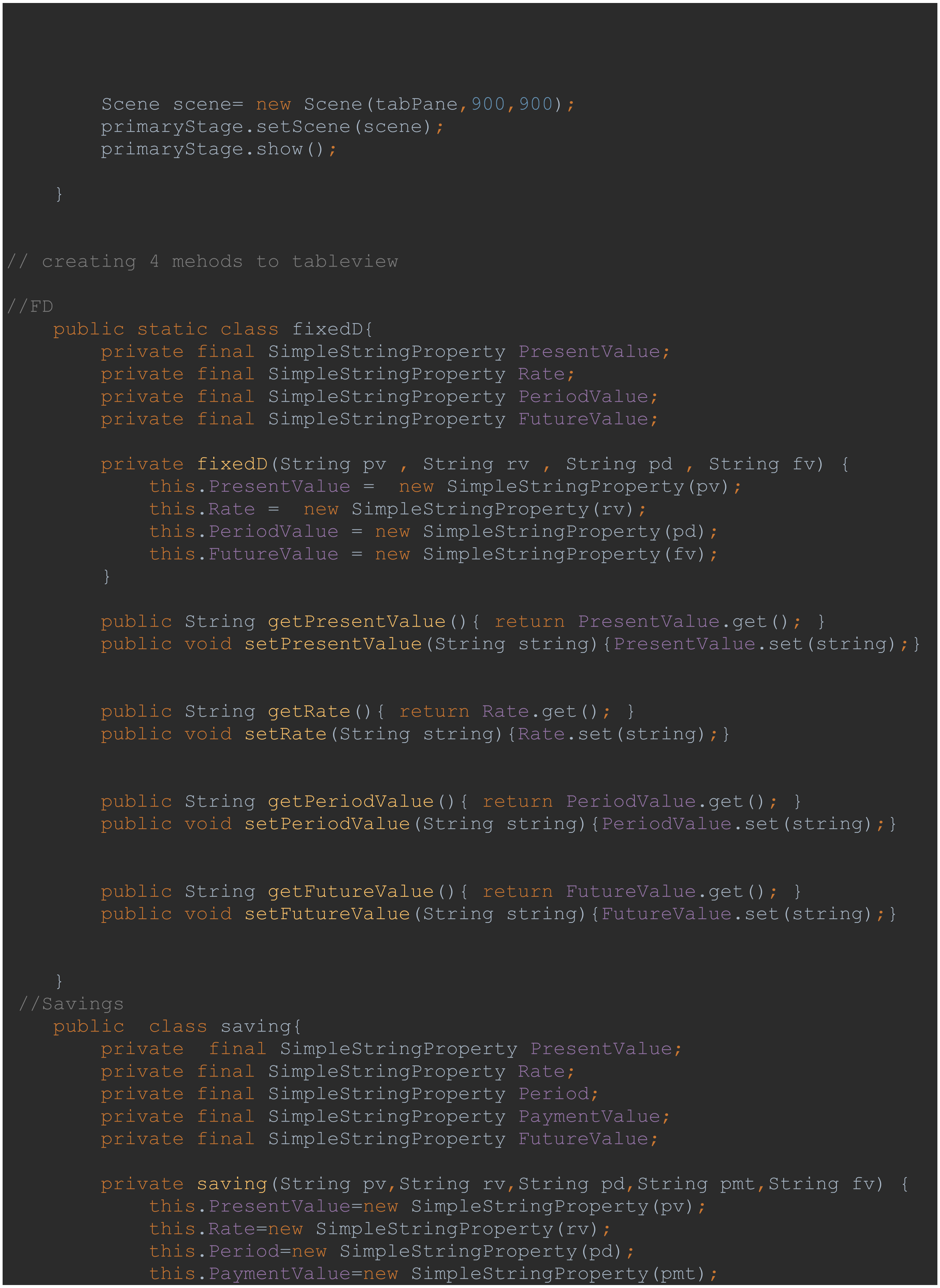
```
                "-fx-font-size: 16;"+
                "-fx-border-color: black;"+
                "-fx-background-color: lavenderblush;"+
                "-fx-border-width: 3");
        gridPane4.add(history4,1,12);

        Pane pane3=new Pane();
        pane3.setStyle("-fx-background-color: white");


        Scene scene4=new Scene(pane3,400,400);
        Stage stage4=new Stage();
        stage4.setTitle("Mortgage History");
        stage4.setScene(scene4);

        history4.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                stage4.show();
            }
        });

        mortgagecalTableView.setEditable(true);

        TableColumn hpColumn=new TableColumn("Loan Value");
        hpColumn.setCellValueFactory(new
PropertyValueFactory<>("LoanValue"));
        TableColumn rColumn3=new TableColumn("Rate");
        rColumn3.setCellValueFactory(new PropertyValueFactory<>("Rate"));
        TableColumn pColumn3=new TableColumn("Loan Term");
        pColumn3.setCellValueFactory(new PropertyValueFactory<>("LoanTerm"));
        TableColumn pmtColumn3=new TableColumn("Payment");
        pmtColumn3.setCellValueFactory(new
PropertyValueFactory<>("Payment"));
        TableColumn down_payment=new TableColumn("Down Payment");
        down_payment.setCellValueFactory(new
PropertyValueFactory<>("DownPayment"));




        mortgagecalTableView.setItems(mortgage1);
        mortgagecalTableView.getColumns().addAll(hpColumn, rColumn3,
pColumn3, pmtColumn3,down_payment);
        pane3.getChildren().add(mortgagecalTableView);

        button4.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                try{
                    if
(comboBox3.getSelectionModel().getSelectedItem().equals("Loan Term")){
                        double r=(Double.parseDouble(tFR3.getText()))/100;
                        double p=Double.parseDouble(tfL.getText());
                        double pmt=Double.parseDouble(tFMPMT.getText());
                        double dpmt=Double.parseDouble(tFDPMT.getText());
                         p=p-dpmt;
```



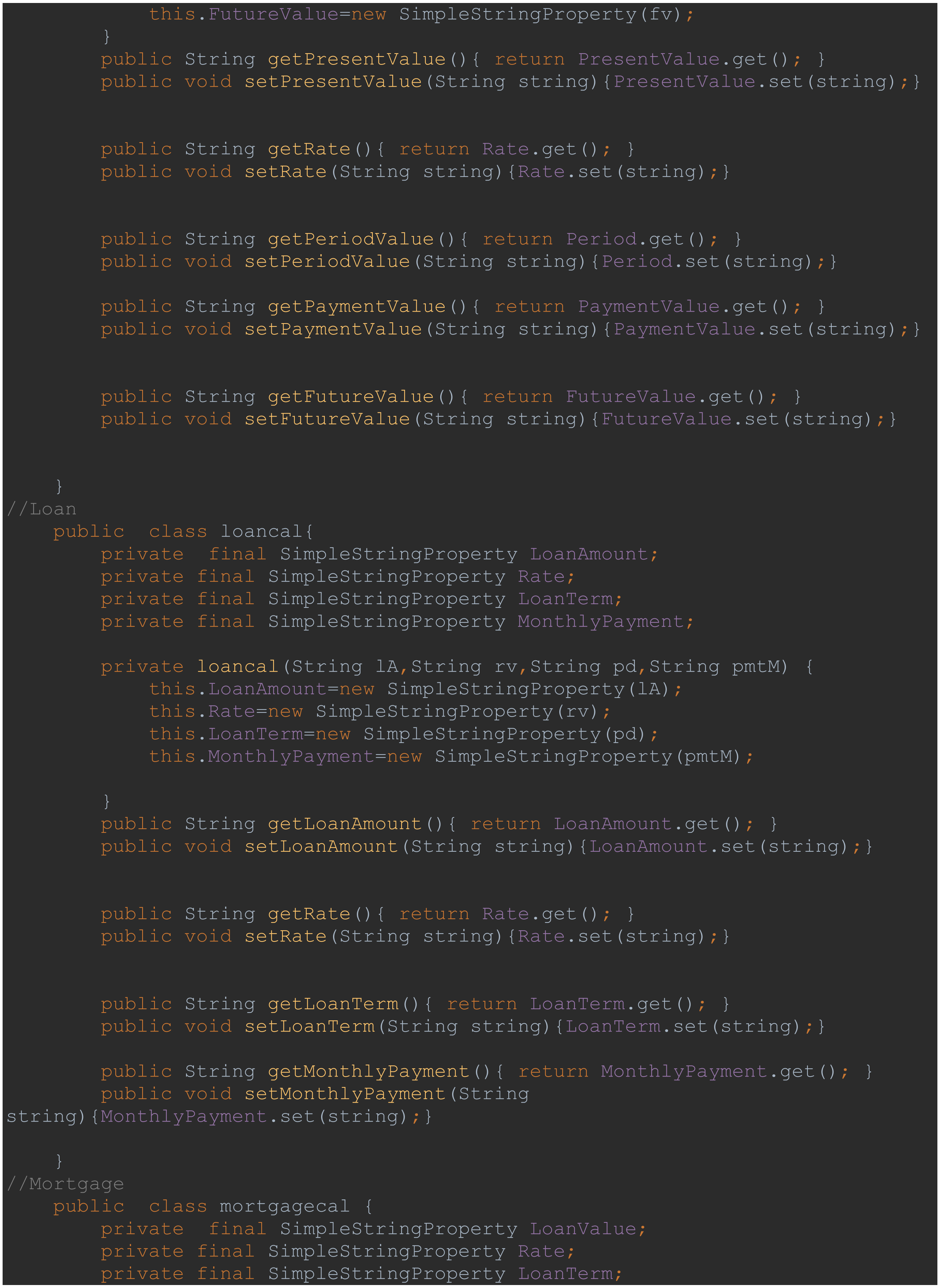
```
                        double a=pmt/(r/12);
                        double b=a-p;
                        double c=Math.log(a/b);
                        double d=Math.log(1+(r/12));
                        double e=c/d;
                        double term=e/12;

tFPD3.setText(String.valueOf(decimalFormat.format(term)));
                    }
                    else
if(comboBox3.getSelectionModel().getSelectedItem().equals("Monthly PMT")){
                        double r=(Double.parseDouble(tFR3.getText()))/100;
                        double p=Double.parseDouble(tfL.getText());
                        double t=Double.parseDouble(tFPD3.getText());
                        double dpmt=Double.parseDouble(tFDPMT.getText());
                        p=p-dpmt;
                        double a=r/12;
                        double b=p*a;
                        double c=1+a;
                        double d=Math.pow(c,(12*t));
                        double e=b*d;
                        double f=d-1;
                        double Mpmt=e/f;

tFMPMT.setText(String.valueOf(decimalFormat.format(Mpmt)));
                    }
                    else
if(comboBox3.getSelectionModel().getSelectedItem().equals("Home Price")){
                        double r=(Double.parseDouble(tFR3.getText()))/100;
                        double t=Double.parseDouble(tFPD3.getText());
                        double dpmt=Double.parseDouble(tFDPMT.getText());
                        double pmt=Double.parseDouble(tFMPMT.getText());
                        double a=r/12;
                        double b=1+a;
                        double c=Math.pow(b,12*t);
                        double d=c-1;
                        double e=pmt*d;
                        double f=a*c;
                        double g=e/f;
                        double Home=dpmt+g;

tfL.setText(String.valueOf(decimalFormat.format(Home)));

                    }
                    else
if(comboBox3.getSelectionModel().getSelectedItem().equals("Down PMT")){
                        double r=(Double.parseDouble(tFR3.getText()))/100;
                        double p=Double.parseDouble(tfL.getText());
                        double t=Double.parseDouble(tFPD3.getText());
                        double pmt=Double.parseDouble(tFMPMT.getText());
                        double a=r/12;
                        double b=1+a;
                        double c=Math.pow(b,12*t);
                        double d=c-1;
                        double e=pmt*d;
                        double f=a*c;
                        double g=e/f;
```



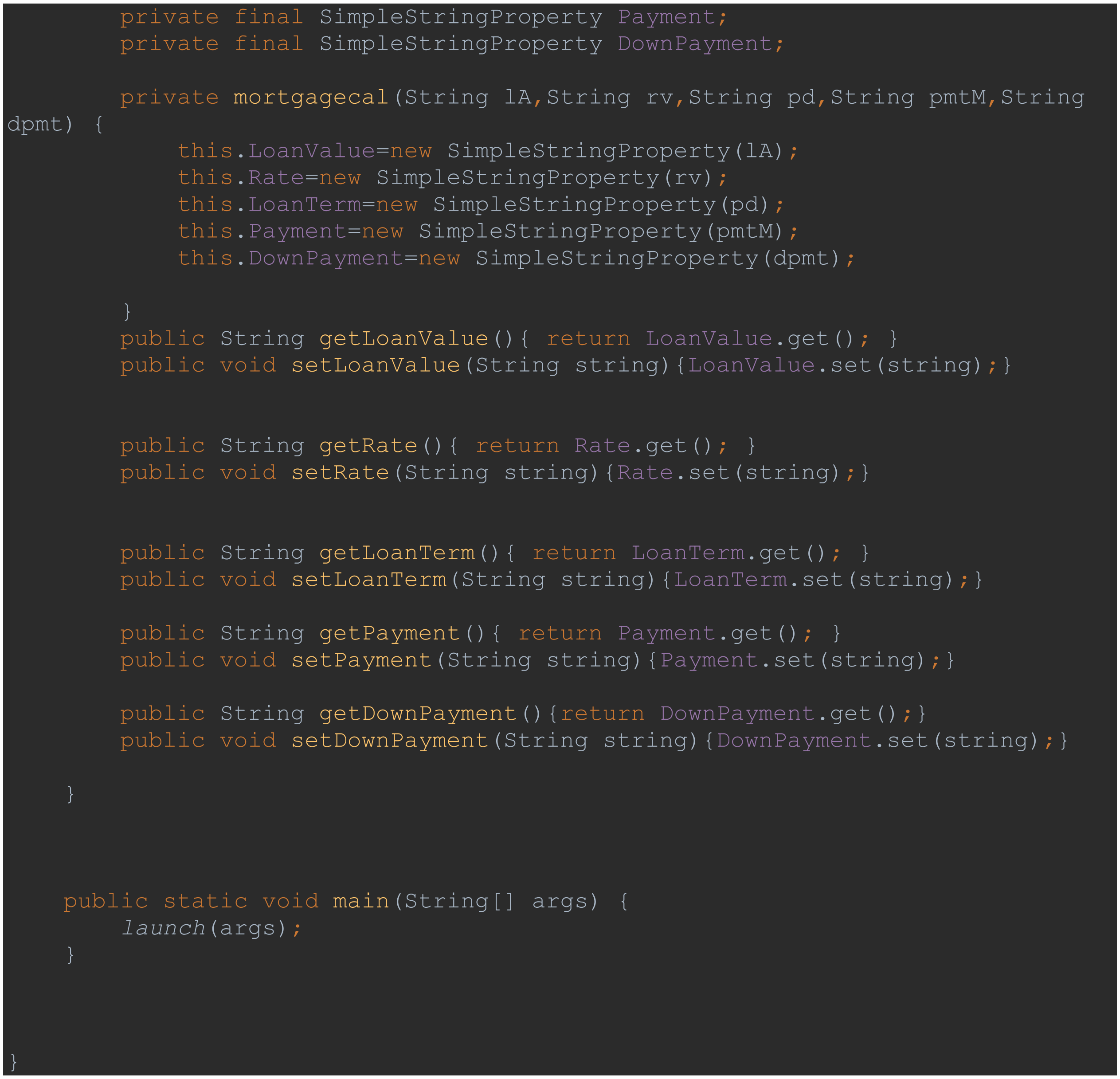
```
                        double Down=p-g;

tFDPMT.setText(String.valueOf(decimalFormat.format(Down)));

                    }
                    mortgage1.add(new mortgagecal(
                            tfL.getText(),
                            tFR3.getText(),
                            tFPD3.getText(),
                            tFMPMT.getText(),
                            tFDPMT.getText()));


                }
                catch (NullPointerException e){
                    System.out.println("Please select From ComboBox");
                }
                catch (NumberFormatException a){
                    System.out.println("Please Type numbers and decimal point
only");
                }



            }
        });

        GridPane gridPane5=new GridPane();
        gridPane5.setVgap(15);
        gridPane5.setHgap(15);
        Tab tab5=new Tab("Help",gridPane5);
        tab5.setStyle("-fx-font-family: 'Arial Black';"+
                "-fx-background-color: #abd7c0");
        tabPane.getTabs().add(tab5);

        Label label5=new Label("In This Financial Calculator you can do
FD,Savings,Loan,Mortgage calculations. \n"+
                "\n"+
                "1) First you need to select calculator from tabs. (eg-Fixed
Deposit)\n"+
                "\n"+
                "2) Next you can choose from combo box wisht to calculate
value (eg-Period) \n"+
                "\n"+
                "3) Enter values to textfields by using numberpad, except the
textfield \n"+
                "  which choose from combo box.\n"+
                "\n"+
                "4) Click 'Calculate' button then you can see the results
also you can \n"+
                "the history by clicking 'History Button'. ");
        label5.setStyle(
                "-fx-font-family: 'Arial Black';"+
                        "-fx-text-fill: dimgray;"+
                        "-fx-font-size: 18");
        gridPane5.add(label5,3,2);
```



```
        Scene scene= new Scene(tabPane,900,900);
        primaryStage.setScene(scene);
        primaryStage.show();

    }


// creating 4 mehods to tableview

//FD
    public static class fixedD{
        private final SimpleStringProperty PresentValue;
        private final SimpleStringProperty Rate;
        private final SimpleStringProperty PeriodValue;
        private final SimpleStringProperty FutureValue;

        private fixedD(String pv , String rv , String pd , String fv) {
            this.PresentValue =  new SimpleStringProperty(pv);
            this.Rate =  new SimpleStringProperty(rv);
            this.PeriodValue = new SimpleStringProperty(pd);
            this.FutureValue = new SimpleStringProperty(fv);
        }

        public String getPresentValue(){ return PresentValue.get(); }
        public void setPresentValue(String string){PresentValue.set(string);}


        public String getRate(){ return Rate.get(); }
        public void setRate(String string){Rate.set(string);}


        public String getPeriodValue(){ return PeriodValue.get(); }
        public void setPeriodValue(String string){PeriodValue.set(string);}


        public String getFutureValue(){ return FutureValue.get(); }
        public void setFutureValue(String string){FutureValue.set(string);}


    }
 //Savings
    public  class saving{
        private  final SimpleStringProperty PresentValue;
        private final SimpleStringProperty Rate;
        private final SimpleStringProperty Period;
        private final SimpleStringProperty PaymentValue;
        private final SimpleStringProperty FutureValue;

        private saving(String pv,String rv,String pd,String pmt,String fv) {
            this.PresentValue=new SimpleStringProperty(pv);
            this.Rate=new SimpleStringProperty(rv);
            this.Period=new SimpleStringProperty(pd);
            this.PaymentValue=new SimpleStringProperty(pmt);
```



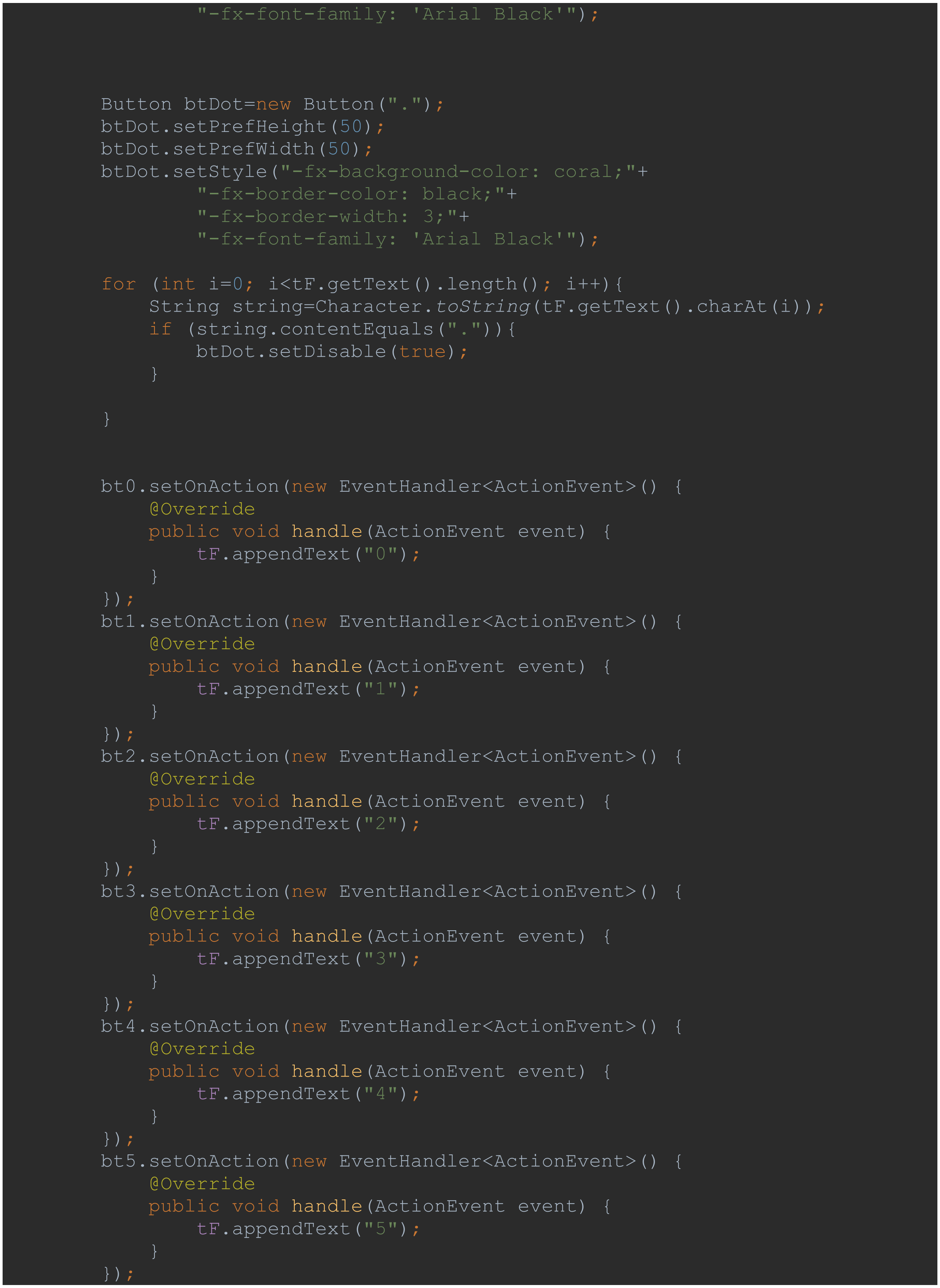
```
            this.FutureValue=new SimpleStringProperty(fv);
        }
        public String getPresentValue(){ return PresentValue.get(); }
        public void setPresentValue(String string){PresentValue.set(string);}


        public String getRate(){ return Rate.get(); }
        public void setRate(String string){Rate.set(string);}


        public String getPeriodValue(){ return Period.get(); }
        public void setPeriodValue(String string){Period.set(string);}

        public String getPaymentValue(){ return PaymentValue.get(); }
        public void setPaymentValue(String string){PaymentValue.set(string);}


        public String getFutureValue(){ return FutureValue.get(); }
        public void setFutureValue(String string){FutureValue.set(string);}


    }
//Loan
    public  class loancal{
        private  final SimpleStringProperty LoanAmount;
        private final SimpleStringProperty Rate;
        private final SimpleStringProperty LoanTerm;
        private final SimpleStringProperty MonthlyPayment;

        private loancal(String lA,String rv,String pd,String pmtM) {
            this.LoanAmount=new SimpleStringProperty(lA);
            this.Rate=new SimpleStringProperty(rv);
            this.LoanTerm=new SimpleStringProperty(pd);
            this.MonthlyPayment=new SimpleStringProperty(pmtM);

        }
        public String getLoanAmount(){ return LoanAmount.get(); }
        public void setLoanAmount(String string){LoanAmount.set(string);}


        public String getRate(){ return Rate.get(); }
        public void setRate(String string){Rate.set(string);}


        public String getLoanTerm(){ return LoanTerm.get(); }
        public void setLoanTerm(String string){LoanTerm.set(string);}

        public String getMonthlyPayment(){ return MonthlyPayment.get(); }
        public void setMonthlyPayment(String
string){MonthlyPayment.set(string);}

    }
//Mortgage
    public  class mortgagecal {
        private  final SimpleStringProperty LoanValue;
        private final SimpleStringProperty Rate;
        private final SimpleStringProperty LoanTerm;
```



```
        private final SimpleStringProperty Payment;
        private final SimpleStringProperty DownPayment;

        private mortgagecal(String lA,String rv,String pd,String pmtM,String 
dpmt) {
            this.LoanValue=new SimpleStringProperty(lA);
            this.Rate=new SimpleStringProperty(rv);
            this.LoanTerm=new SimpleStringProperty(pd);
            this.Payment=new SimpleStringProperty(pmtM);
            this.DownPayment=new SimpleStringProperty(dpmt);

        }
        public String getLoanValue(){ return LoanValue.get(); }
        public void setLoanValue(String string){LoanValue.set(string);}


        public String getRate(){ return Rate.get(); }
        public void setRate(String string){Rate.set(string);}


        public String getLoanTerm(){ return LoanTerm.get(); }
        public void setLoanTerm(String string){LoanTerm.set(string);}

        public String getPayment(){ return Payment.get(); }
        public void setPayment(String string){Payment.set(string);}

        public String getDownPayment(){return DownPayment.get();}
        public void setDownPayment(String string){DownPayment.set(string);}

    }



    public static void main(String[] args) {
        launch(args);
    }



}
```

numberPad1.java

```java
package sample;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;

public class numberPad1 {
    // creating method to set all calculators in Main.java
    public static void numberPad(TextField tF, GridPane gP){
        //creating buttons and add them to gridpane
        Button bt0=new Button("0");
        bt0.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");
        bt0.setPrefHeight(50);
        bt0.setPrefWidth(50);


        Button bt1=new Button("1");
        bt1.setPrefHeight(50);
        bt1.setPrefWidth(50);
        bt1.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");



        Button bt2=new Button("2");
        bt2.setPrefHeight(50);
        bt2.setPrefWidth(50);
        bt2.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt3=new Button("3");
        bt3.setPrefHeight(50);
        bt3.setPrefWidth(50);
        bt3.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt4=new Button("4");
        bt4.setPrefHeight(50);
        bt4.setPrefWidth(50);
        bt4.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt5=new Button("5");
        bt5.setPrefHeight(50);
        bt5.setPrefWidth(50);
        bt5.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt6=new Button("6");
        bt6.setPrefHeight(50);
        bt6.setPrefWidth(50);
        bt6.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt7=new Button("7");
        bt7.setPrefHeight(50);
        bt7.setPrefWidth(50);
        bt7.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt8=new Button("8");
        bt8.setPrefHeight(50);
        bt8.setPrefWidth(50);
        bt8.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button bt9=new Button("9");
        bt9.setPrefHeight(50);
        bt9.setPrefWidth(50);
        bt9.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
                "-fx-font-family: 'Arial Black'");


        Button btC=new Button("C");
        btC.setPrefHeight(50);
        btC.setPrefWidth(50);
        btC.setStyle("-fx-background-color: coral;"+
                "-fx-border-color: black;"+
                "-fx-border-width: 3;"+
```



```
        "-fx-font-family: 'Arial Black'");



Button btDot=new Button(".");
btDot.setPrefHeight(50);
btDot.setPrefWidth(50);
btDot.setStyle("-fx-background-color: coral;"+
        "-fx-border-color: black;"+
        "-fx-border-width: 3;"+
        "-fx-font-family: 'Arial Black'");

for (int i=0; i<tF.getText().length(); i++){
    String string=Character.toString(tF.getText().charAt(i));
    if (string.contentEquals(".")){
        btDot.setDisable(true);
    }

}


bt0.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("0");
    }
});
bt1.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("1");
    }
});
bt2.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("2");
    }
});
bt3.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("3");
    }
});
bt4.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("4");
    }
});
bt5.setOnAction(new EventHandler<ActionEvent>() {
    @Override
    public void handle(ActionEvent event) {
        tF.appendText("5");
    }
});
```

```
        bt6.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.appendText("6");
            }
        });
        bt7.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.appendText("7");
            }
        });
        bt8.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.appendText("8");
            }
        });
        bt9.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.appendText("9");
            }
        });
        btC.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.setText(tF.getText().substring(0,tF.getText().length()-
1));

                btDot.setDisable(false);
            }
        });
        btDot.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                tF.appendText(".");
                btDot.setDisable(true);
                for (int i=0; i<tF.getText().length(); i++){
                    String string=Character.toString(tF.getText().charAt(i));
                    if (string.contentEquals(".")){
                        btDot.setDisable(true);
                    }

                }
            }
        });

//adding numberpad to gridpane
        gP.add(bt0,4,18);
        gP.add(bt1,3,15);
        gP.add(bt2,4,15);
        gP.add(bt3,5,15);
        gP.add(bt4,3,16);
        gP.add(bt5,4,16);
        gP.add(bt6,5,16);
        gP.add(bt7,3,17);
        gP.add(bt8,4,17);
        gP.add(bt9,5,17);
        gP.add(btC,3,18);
        gP.add(btDot,5,18);

    }

    }
```