**import** javafx.application.Application;

**import** javafx.event.ActionEvent;

**import** javafx.event.EventHandler;

**import** javafx.scene.Scene;

**import** javafx.scene.control.Button;

**import** javafx.scene.control.TextArea;

**import** javafx.scene.layout.HBox;

**import** javafx.scene.layout.VBox;

**import** javafx.stage.Stage;

/\*\*

\* Calculator application

\* text file encoding utf-8

\*/

**public** **class** Calculator **extends** Application {

/\*\*

\* declare of objects and variables that will be use

\*/

TextArea textarea = **new** TextArea();

String showExpression = "";

String hiddenExpression = "";

**boolean** isSquare = **false**;

**boolean** isOpen = **false**;

**boolean** isequal = **false**;

**int** countOpen = 0;

**static** String *answer*;

**int** countClose = 0;

**long** startTime;

**long** currentTime;

VBox vbox2; //design pane

@Override

**public** **void** start(Stage arg0) **throws** Exception {

/\*\*

\* call function of calculator design that contain all design of the

\* calculator and return v box that contain all design of the calculator

\*/

vbox2 = CalculatorUI();

Scene scene = **new** Scene(vbox2, 350, 400);

arg0.setTitle("Calcluator");

arg0.setScene(scene);

arg0.show();

arg0.setOnCloseRequest(e -> {

arg0.close();

});

}

/\*\*

\* function called CalculatorUI that contains design of the calculator and include

\* v box contain text area and 3 h boxes (h box1,h box2, and h box3 ) h

\* box 1 contains buttons 1 , 2 ,3 , clear , and reset h box 2 contains

\* buttons 4 , 5 ,6 , multiplication and division h box 3 contains

\* buttons 7 , 8 ,9 and brackets h box 4 contains buttons 0 , . ,%,+ and

\* - h box 5 contains buttons x^2 ,root and =

\* and set the width and height of buttons and variables and objects to be

\* interactive with the width and height of the window if user minimize

\* or maximize window

\* and action of buttons and components

\* **@return**

\*/

**public** VBox CalculatorUI() {

VBox vbox = **new** VBox();

Button one = **new** Button("1"); //buttons

Button two = **new** Button("2");

Button three = **new** Button("3");

Button four = **new** Button("4");

Button five = **new** Button("5");

Button six = **new** Button("6");

Button seven = **new** Button("7");

Button eight = **new** Button("8");

Button nine = **new** Button("9");

Button plus = **new** Button("+");

Button minus = **new** Button("-");

Button mul = **new** Button("\*");

Button div = **new** Button("/");

Button sqrt = **new** Button("2√");

Button mod = **new** Button("%");

Button dot = **new** Button(".");

Button square = **new** Button("x^2");

Button equal = **new** Button("=");

Button open = **new** Button("(");

Button close = **new** Button(")");

Button delete = **new** Button("Del");

Button zero = **new** Button("0");

Button clear = **new** Button("Clear");

Button sin = **new** Button("sin");

Button cos = **new** Button("cos");

Button tan = **new** Button("tan");

Button ans = **new** Button("ans");

Button pi = **new** Button("pi");

HBox hbox1 = **new** HBox();

hbox1.getChildren().addAll(one, two, three, delete, clear);

HBox hbox2 = **new** HBox();

hbox2.getChildren().addAll(four, five, six, mul, div);

HBox hbox3 = **new** HBox();

hbox3.getChildren().addAll(seven, eight, nine, open, close);

HBox hbox4 = **new** HBox();

hbox4.getChildren().addAll(zero, dot, mod,plus, minus);

HBox hbox5 = **new** HBox();

hbox5.getChildren().addAll(square, sqrt,sin,cos,tan);

HBox hbox6=**new** HBox();

hbox6.getChildren().addAll(pi,ans,equal);

sin.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

sin.prefHeightProperty().bind(hbox5.heightProperty());

cos.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

cos.prefHeightProperty().bind(hbox5.heightProperty());

tan.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

tan.prefHeightProperty().bind(hbox5.heightProperty());

ans.prefWidthProperty().bind(hbox6.widthProperty().divide(5));

ans.prefHeightProperty().bind(hbox6.heightProperty());

pi.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

pi.prefHeightProperty().bind(hbox5.heightProperty());

one.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

one.prefHeightProperty().bind(hbox1.heightProperty());

two.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

two.prefHeightProperty().bind(hbox1.heightProperty());

three.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

three.prefHeightProperty().bind(hbox1.heightProperty());

four.prefWidthProperty().bind(hbox2.widthProperty().divide(5));

four.prefHeightProperty().bind(hbox1.heightProperty());

five.prefWidthProperty().bind(hbox2.widthProperty().divide(5));

five.prefHeightProperty().bind(hbox1.heightProperty());

six.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

six.prefHeightProperty().bind(hbox1.heightProperty());

seven.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

seven.prefHeightProperty().bind(hbox1.heightProperty());

eight.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

eight.prefHeightProperty().bind(hbox1.heightProperty());

nine.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

nine.prefHeightProperty().bind(hbox1.heightProperty());

zero.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

zero.prefHeightProperty().bind(hbox1.heightProperty());

plus.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

plus.prefHeightProperty().bind(hbox1.heightProperty());

minus.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

minus.prefHeightProperty().bind(hbox1.heightProperty());

mul.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

mul.prefHeightProperty().bind(hbox1.heightProperty());

div.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

div.prefHeightProperty().bind(hbox1.heightProperty());

dot.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

dot.prefHeightProperty().bind(hbox1.heightProperty());

mod.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

mod.prefHeightProperty().bind(hbox1.heightProperty());

open.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

open.prefHeightProperty().bind(hbox1.heightProperty());

close.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

close.prefHeightProperty().bind(hbox1.heightProperty());

delete.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

delete.prefHeightProperty().bind(hbox1.heightProperty());

clear.prefWidthProperty().bind(hbox1.widthProperty().divide(5));

clear.prefHeightProperty().bind(hbox1.heightProperty());

square.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

square.prefHeightProperty().bind(hbox5.heightProperty());

sqrt.prefWidthProperty().bind(hbox5.widthProperty().divide(5));

sqrt.prefHeightProperty().bind(hbox5.heightProperty());

equal.prefHeightProperty().bind(hbox5.heightProperty());

equal.prefWidthProperty().bind(hbox6.widthProperty().divide(5).multiply(3));

vbox.getChildren().addAll(textarea, hbox1, hbox2, hbox3, hbox4, hbox5,hbox6);

hbox1.prefWidthProperty().bind(vbox.widthProperty());

hbox2.prefWidthProperty().bind(vbox.widthProperty());

hbox3.prefWidthProperty().bind(vbox.widthProperty());

hbox4.prefWidthProperty().bind(vbox.widthProperty());

hbox5.prefWidthProperty().bind(vbox.widthProperty());

hbox1.prefHeightProperty().bind(vbox.heightProperty().divide(7));

hbox2.prefHeightProperty().bind(vbox.heightProperty().divide(7));

hbox3.prefHeightProperty().bind(vbox.heightProperty().divide(7));

hbox4.prefHeightProperty().bind(vbox.heightProperty().divide(7));

hbox5.prefHeightProperty().bind(vbox.heightProperty().divide(7));

hbox6.prefHeightProperty().bind(vbox.heightProperty().divide(7));

textarea.prefWidthProperty().bind(vbox.widthProperty());

textarea.prefHeightProperty().bind(vbox.heightProperty().divide(7));

//sin button

sin.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if**(ch=='.'){

showExpression+="0";

hiddenExpression+="0";

ch='0';

}

**if**(showExpression.equals("")){

showExpression+="sin(";

hiddenExpression+="@(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

**else** **if**(Character.*isDigit*(ch)|| ch==')'|| isSquare|| ch=='i'){

showExpression+="\*sin(";

hiddenExpression+="\*@(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

**else** **if**(ch=='('|| ch=='\*' || ch=='/' || ch=='+' || ch=='-' || ch=='%'||ch=='√'){

showExpression+="sin(";

hiddenExpression+="@(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

});

//cos button

cos.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if**(ch=='.'){

showExpression+="0";

hiddenExpression+="0";

ch='0';

}

**if**(showExpression.equals("")){

showExpression+="cos(";

hiddenExpression+="1\*&(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

**else** **if**(Character.*isDigit*(ch)|| ch==')'|| isSquare|| ch=='i'){

showExpression+="\*cos(";

hiddenExpression+="\*&(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

**else** **if**(ch=='('|| ch=='\*' || ch=='/' || ch=='+' || ch=='-' || ch=='%' ||ch=='√'){

showExpression+="cos(";

hiddenExpression+="&(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

});

//tan button

tan.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if**(ch=='.'){

showExpression+="0";

hiddenExpression+="0";

ch='0';

}

**if**(showExpression.equals("")){

showExpression+="tan(";

hiddenExpression+="1\*#(";

textarea.setText(showExpression);

isOpen=**true**;

countOpen++;

isequal=**false**;

}

**else** **if**(Character.*isDigit*(ch)|| ch==')'|| isSquare|| ch=='i'){

showExpression+="\*tan(";

hiddenExpression+="\*#(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

**else** **if**(ch=='('|| ch=='\*' || ch=='/' || ch=='+' || ch=='-' || ch=='%' ||ch=='√'){

showExpression+="tan(";

hiddenExpression+="#(";

textarea.setText(showExpression);

countOpen++;

isOpen=**true**;

isequal=**false**;

}

});

//pi button

pi.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if**(isequal){

showExpression="pi";

hiddenExpression="1\*3.1428";

textarea.setText(showExpression);

isequal=**false**;

}

**if**(showExpression.equals("")){

showExpression+="pi";

hiddenExpression+="1\*3.1428";

textarea.setText(showExpression);

}

**else**{

showExpression+="pi";

hiddenExpression+="1\*3.1428";

textarea.setText(showExpression);

}

});

//button one

/\*\*

\* when click on button 1 we check if last character is not ')' and is

\* square is false set 1 to show expression and set 1 to hidden

\* expression that we calculate later

\*/

one.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "1";

hiddenExpression += "1";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "1";

hiddenExpression = "1";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button two

/\*\*

\* when click on button 2 we check if last character is not ')' and is

\* square is false set 2 to show expression and set 2 to hidden

\* expression that we calculate later

\*/

two.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "2";

hiddenExpression += "2";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "2";

hiddenExpression = "2";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button three

/\*\*

\* when click on button 3 we check if last character is not ')' and is

\* square is false set 3 to show expression and set 3 to hidden

\* expression that we calculate later

\*/

three.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "3";

hiddenExpression += "3";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "3";

hiddenExpression = "3";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button four

/\*\*

\* when click on button 4 we check if last character is not ')' and is

\* square is false set 4 to show expression and set 4 to hidden

\* expression that we calculate later

\*/

four.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "4";

hiddenExpression += "4";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "4";

hiddenExpression = "4";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button five

/\*\*

\* when click on button 5 we check if last character is not ')' and is

\* square is false set 5 to show expression and set 5 to hidden

\* expression that we calculate later

\*/

five.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "5";

hiddenExpression += "5";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "5";

hiddenExpression = "5";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button six

/\*\*

\* when click on button 6 we check if last character is not ')' and is

\* square is false set 6 to show expression and set 6 to hidden

\* expression that we calculate later

\*/

six.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "6";

hiddenExpression += "6";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "6";

hiddenExpression = "6";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button seven

/\*\*

\* when click on button 7 we check if last character is not ')' and is

\* square is false set 7 to show expression and set 6 to hidden

\* expression that we calculate later

\*/

seven.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "7";

hiddenExpression += "7";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "7";

hiddenExpression = "7";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button eight

/\*\*

\* when click on button 8 we check if last character is not ')' and is

\* square is false set 8 to show expression and set 8 to hidden

\* expression that we calculate later

\*/

eight.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "8";

hiddenExpression += "8";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "8";

hiddenExpression = "8";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button nine

/\*\*

\* when click on button 9 we check if last character is not ')' and is

\* square is false set 9 to show expression and set 9 to hidden

\* expression that we calculate later

\*/

nine.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "9";

hiddenExpression += "9";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "9";

hiddenExpression = "9";

textarea.setText(showExpression);

isequal = **false**;

}

});

//button zero

/\*\*

\* when click on button 0 we check if last character is not ')' and is

\* not / and is not % because we can't divide by zero and is square is

\* false set 0 to show expression and set 0 to hidden expression that we

\* calculate later

\*/

zero.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false**)) {

showExpression += "0";

hiddenExpression += "0";

textarea.setText(showExpression);

}

**if** (isequal) {

showExpression = "0";

hiddenExpression = "0";

textarea.setText(showExpression);

isequal = **false**;

}

});

//dot button

/\*\*

\* when click on button . we check if last character is not ')' and is

\* square is false set 0 to show expression and set 0 to hidden

\* expression that we calculate later

\*/

dot.setOnAction(e -> {

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** ((ch != ')' && isSquare == **false** && Character.*isDigit*(ch))) {

showExpression += ".";

hiddenExpression += ".";

textarea.setText(showExpression);

}

**else** **if**(isequal||showExpression.equals("")){

showExpression = "0.";

hiddenExpression = "0.";

textarea.setText(showExpression);

isequal=**false**;

}

**else**{

showExpression += "0.";

hiddenExpression += "0.";

textarea.setText(showExpression);

isequal=**false**;

}

});

//plus button

/\*\*

\* when click on button + we check set + to show expression and set + to

\* hidden expression that we calculate later

\*/

plus.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try**{

**char** ch = ' ';

ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if**(isequal){

**if**(showExpression.equals("error expression")){

showExpression = "";

hiddenExpression = "";

isSquare=**false**;

}

**else**{

showExpression += "+";

hiddenExpression += "+";

textarea.setText(showExpression);

isequal=**false**;

isSquare=**false**;

}

}

**else** **if**(ch==')'||ch=='.'){

isSquare=**false**;

showExpression += "+";

hiddenExpression += "+";

textarea.setText(showExpression);

}

**else** **if** (!Character.*isDigit*(ch)){}

**else**{

showExpression += "+";

hiddenExpression += "+";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='i'){

showExpression += "+";

hiddenExpression += "+";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch**(Exception e){}

}

});

//minus button

/\*\*

\* when click on button - we check set - to show expression and set - to

\* hidden expression that we calculate later

\*/

minus.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

**char** ch = ' ';

**char** ch1 = ' ';

**if**(!showExpression.equals("")){

ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if**(ch=='-'){

ch1 = showExpression.charAt(showExpression.length() - 2);

**if**(ch1=='-'){

showExpression = "error expression";

hiddenExpression = "error expression";

textarea.setText(showExpression);

isequal=**true**;

}

}

}

**if** (isequal) {

**if** (showExpression.equals("error expression")) {

showExpression = "";

hiddenExpression = "";

}

**else** {

showExpression += "-";

hiddenExpression += "-";

textarea.setText(showExpression);

isequal = **false**;

}

}

**else** **if** (showExpression == "") {

showExpression += "-";

hiddenExpression += "-";

textarea.setText(showExpression);

}

**else** {

showExpression += "-";

hiddenExpression += "-";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch** (Exception e) {}

}

});

//div button

/\*\*

\* action of button / when we clicked on this is what be happen

\*/

div.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

isequal = **false**;

**char** ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if** (isequal) {

**if** (showExpression.equals("error expression")) {

showExpression = "";

hiddenExpression = "";

}

**else** {

showExpression += "/";

hiddenExpression += "/";

textarea.setText(showExpression);

isequal = **false**;

}

}

**else** **if** (Character.*isDigit*(ch) || ch == ')') {

showExpression += "/";

hiddenExpression += "/";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='.'){

showExpression += "/";

hiddenExpression += "/";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='i'){

showExpression += "/";

hiddenExpression += "/";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch** (Exception e) {}

}

});

//mulplication button

/\*\*

\* action of button \* when we clicked on this is what be happen

\*/

mul.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

isequal = **false**;

**char** ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if** (isequal) {

**if** (showExpression.equals("error expression")) {

showExpression = "";

hiddenExpression = "";

}

**else** {

showExpression += "\*";

hiddenExpression += "\*";

textarea.setText(showExpression);

isequal = **false**;

}

}

**else** **if** (Character.*isDigit*(ch) || ch == ')') {

showExpression += "\*";

hiddenExpression += "\*";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='.'){

showExpression += "\*";

hiddenExpression += "\*";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='i'){

showExpression += "\*";

hiddenExpression += "\*";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch** (Exception e) {}

}

});

//mod button

/\*\*

\* action of button % when we clicked on this is what be happen

\*/

mod.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

isequal = **false**;

**char** ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if** (isequal) {

**if** (showExpression.equals("error expression")) {

showExpression = "";

hiddenExpression = "";

}

**else** {

showExpression += "%";

hiddenExpression += "%";

textarea.setText(showExpression);

isequal = **false**;

}

}

**else** **if** (Character.*isDigit*(ch) || ch == ')') {

showExpression += "%";

hiddenExpression += "%";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='.'){

showExpression += "%";

hiddenExpression += "%";

textarea.setText(showExpression);

isSquare = **false**;

}

**if**(ch=='i'){

showExpression += "%";

hiddenExpression += "%";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch** (Exception e) {}

}

});

//sqrt button

/\*\*

\* action of button '2√' when we clicked on this is what be happen

\*/

sqrt.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

isequal = **false**;

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if**(ch=='.'){

hiddenExpression+='0';

ch='0';

}

**if** (showExpression.equals("")) {

showExpression += "√";

hiddenExpression += "|";

textarea.setText(showExpression);

isequal=**false**;

}

**else** **if** (!Character.*isDigit*(ch) && isSquare == **false** && ch != ')'&& ch != '?') {

showExpression += "√";

hiddenExpression += "|";

textarea.setText(showExpression);

isequal=**false**;

}

**if** (Character.*isDigit*(ch)){

showExpression += "√";

hiddenExpression += "\*|";

textarea.setText(showExpression);

}

**if**(ch=='i'){

showExpression += "√";

hiddenExpression += "√";

textarea.setText(showExpression);

isSquare = **false**;

}

}

**catch** (Exception e) {}

}

});

textarea.setEditable(**false**);

// square button

/\*\*

\* action of button ^2 when we clicked on this is what be happen

\*/

square.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

isequal = **false**;

**char** lastChar=' ';

**int** index=0;

**if**(!showExpression.equals("")){

lastChar = showExpression.charAt(showExpression.length() - 1);

}

**if**(lastChar=='.'){

hiddenExpression+='0';

}

**if** (Character.*isDigit*(lastChar)) {

String number = "";

String swapNumber = "";

String str = hiddenExpression;

**for** (**int** i = str.length()-1; i >= 0; i--) {

**char** ch = str.charAt(i);

**if** (Character.*isDigit*(ch)) {

number += ch;

}

**else** **if**( ch=='.'){

number+=".";

}

**else** {

index=i+1;

**break**;

}

}

**for** (**int** i = number.length()-1; i >= 0; i--) {

**char** ch = number.charAt(i);

swapNumber += ch;

}

**double** temp=Double.*parseDouble*(swapNumber)\*Double.*parseDouble*(swapNumber);

hiddenExpression=hiddenExpression.substring(0, index);

hiddenExpression+=temp;

showExpression += "^2";

textarea.setText(showExpression);

isSquare = **true**;

}

}

**catch** (Exception e) {

showExpression="";

hiddenExpression="";

textarea.setText("error");

}

}

});

//delete button

/\*\*

\* action of button delete when we clicked on this is what be happen

\* that delete last character in showExpression and hiddenExpression to

\* begin again

\*/

delete.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**try** {

**if** (showExpression.equals("error expression")) {

textarea.clear();

}

**char** ch = ' ';

**if** (!showExpression.equals("")) {

ch = showExpression.charAt(showExpression.length() - 1);

}

**if** (ch == ')') {

countClose--;

}

**else** **if** (ch == '(') {

countOpen--;

**if** (countOpen == 0);

isOpen = **false**;

}

isequal=**false**;

String str = textarea.getText();

String newStr = str.substring(0, showExpression.length() - 1);

showExpression = newStr;

hiddenExpression = hiddenExpression.substring(0, hiddenExpression.length() - 1);

textarea.setText(showExpression);

}

**catch** (Exception e) {

showExpression="";

hiddenExpression="";

textarea.setText("error");

}

}

});

//answer button

/\*\*

\*action of button answer when we clicked on this is what be happen

\* that copy last character in showExpression and hiddenExpression to

\* begin again

\*/

ans.setOnAction(e -> {

**if**(!isequal){

showExpression+=*answer*;

hiddenExpression+=*answer*;

textarea.setText(showExpression);

isequal=**false**;

}

});

//reset button

/\*\*

\* action of button reset when we clicked on this is what be happen that

\* clear showExpression and hiddenExpression to begin again

\*/

clear.setOnAction(e -> {

textarea.setText("");

showExpression = "";

hiddenExpression = "";

isOpen = **false**;

isSquare = **false**;

countClose = 0;

countOpen = 0;

});

// open bracket button

/\*\*

\* action of open bracket button is to open another bracket

\*/

open.setOnAction(e -> {

**char** ch1 = ' ';

isequal=**false**;

**if** (showExpression.equals("error expression")) {

showExpression = "";

hiddenExpression = "";

}

**if** (!showExpression.equals("")) {

ch1 = showExpression.charAt(showExpression.length() - 1);

**if**(ch1=='.'){

hiddenExpression+='0';

showExpression+='0';

ch1='0';

}

}

**if** (((!Character.*isDigit*(ch1) && isSquare == **false**) && ch1 != ')') || ch1 == ' ') {

showExpression += "(";

hiddenExpression += "(";

textarea.setText(showExpression);

isOpen = **true**;

countOpen++;

}

**if**(Character.*isDigit*(ch1)){

showExpression += "\*(";

hiddenExpression += "\*(";

textarea.setText(showExpression);

isOpen = **true**;

countOpen++;

}

});

//close bracket button

/\*\*

\* action of close brackets button is to close and not to close if another bracket not

\* open

\*/

close.setOnAction(e -> {

**char** ch1 = ' ';

**if** (!showExpression.equals("")) {

ch1 = showExpression.charAt(showExpression.length() - 1);

**if**(ch1=='.'){

hiddenExpression+="0";

ch1='0';

}

}

**if** (ch1=='(') {

textarea.setText("error expression");

showExpression = "";

hiddenExpression = "";

isOpen=**false**;

}

**if** (isOpen) {

**if**(ch1=='.'){

showExpression += ")";

hiddenExpression += ")";

textarea.setText(showExpression);

countClose++;

}

**else**{

showExpression += ")";

hiddenExpression += ")";

textarea.setText(showExpression);

countClose++;

}

}

});

//equal button

/\*\*

\* action of equal button when clicked on evaluate arithmetic expression

\* that user enter in text area

\*/

equal.setOnAction(**new** EventHandler<ActionEvent>() {

@Override

**public** **void** handle(ActionEvent event) {

**char** ch=' ';

isequal=**true**;

**try** {

**if** (countClose != countOpen) {

textarea.setText("error expression");

showExpression = "";

hiddenExpression = "";

}

**else** {

ch = showExpression.charAt(showExpression.length() - 1);

**if**(ch=='.'){

hiddenExpression+='0';

}

EvaluateExpression ev = **new** EvaluateExpression();

hiddenExpression = ev.brackets(hiddenExpression);

*answer*=hiddenExpression;

textarea.setText(hiddenExpression);

showExpression = hiddenExpression;

}

}

**catch** (Exception e) {

showExpression="";

hiddenExpression="";

textarea.setText("error");

}

}

});

**return** vbox;

}

/\*\*

\* main method that run application

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

Application.*launch*(args);

}

}