import java.awt.BorderLayout;

import java.awt.Color;

import java.awt.GridLayout;

import java.awt.LayoutManager;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JDialog;

import javax.swing.JFrame;

import javax.swing.JPanel;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.WindowConstants;

public class Calculator {

JFrame f=new JFrame("Calculator");

JPanel p1=new JPanel();

JPanel p2=new JPanel(new GridLayout(6,5));

JButton b1=new JButton("三角学");

JButton b2=new JButton("lg");

JButton b3=new JButton("ln");

JButton b4=new JButton("x^2");

JButton b5=new JButton("mod");

JButton b6=new JButton("sqrt");

JButton b7=new JButton("c");

JButton b8=new JButton("←");

JButton b9=new JButton("%");

JButton b10=new JButton("÷");

JButton b11=new JButton("x!");

JButton b12=new JButton("7");

JButton b13=new JButton("8");

JButton b14=new JButton("9");

JButton b15=new JButton("×");

JButton b16=new JButton("1/x");

JButton b17=new JButton("4");

JButton b18=new JButton("5");

JButton b19=new JButton("6");

JButton b20=new JButton("-");

JButton b21=new JButton("x^y");

JButton b22=new JButton("1");

JButton b23=new JButton("2");

JButton b24=new JButton("3");

JButton b25=new JButton("+");

JButton b26=new JButton("π");

JButton b27=new JButton("e");

JButton b28=new JButton("0");

JButton b29=new JButton(".");

JButton b30=new JButton("=");

JButton bj1=new JButton("sinx");

JButton bj2=new JButton("cosx");

JButton bj3=new JButton("tanx");

JButton bj4=new JButton("asinx");

JButton bj5=new JButton("acosx");

JButton bj6=new JButton("atanx");

JButton bj7=new JButton("角度制");

JButton bj8=new JButton("弧度制");

boolean numflag1=false,numflag2=false,opflag=false;

boolean result=false,zero=false,degree=false,addnum=false;

double num1,num2;

int op;

String opstr;

JTextField tf=new JTextField("0",50);

JTextArea ta=new JTextArea(100,15);

JScrollPane sp=new JScrollPane(ta);

public Calculator() {

p1.add(tf);

p2.setBackground(Color.white);

f.add(sp,BorderLayout.EAST);

f.add(p1,BorderLayout.NORTH);

f.add(p2);

p2.add(b1);

p2.add(b2);

p2.add(b3);

p2.add(b4);

p2.add(b5);

p2.add(b6);

p2.add(b7);

p2.add(b8);

p2.add(b9);

p2.add(b10);

p2.add(b11);

p2.add(b12);

p2.add(b13);

p2.add(b14);

p2.add(b15);

p2.add(b16);

p2.add(b17);

p2.add(b18);

p2.add(b19);

p2.add(b20);

p2.add(b21);

p2.add(b22);

p2.add(b23);

p2.add(b24);

p2.add(b25);

p2.add(b26);

p2.add(b27);

p2.add(b28);

p2.add(b29);

p2.add(b30);

b7.setBackground(Color.orange);

b8.setBackground(Color.orange);

b9.setBackground(Color.orange);

b10.setBackground(Color.orange);

b15.setBackground(Color.orange);

b20.setBackground(Color.orange);

b25.setBackground(Color.orange);

b1.setBackground(Color.blue);

b12.setBackground(Color.white);

b13.setBackground(Color.white);

b14.setBackground(Color.white);

b17.setBackground(Color.white);

b18.setBackground(Color.white);

b19.setBackground(Color.white);

b22.setBackground(Color.white);

b23.setBackground(Color.white);

b24.setBackground(Color.white);

b28.setBackground(Color.white);

b30.setBackground(Color.yellow);

bj4.setBackground(Color.white);

bj5.setBackground(Color.white);

bj6.setBackground(Color.white);

bj7.setBackground(Color.LIGHT\_GRAY);

bj8.setBackground(Color.white);

(b2).addActionListener(new signs());

(b3).addActionListener(new signs());

(b4).addActionListener(new signs());

(b6).addActionListener(new signs());

(b9).addActionListener(new signs());

(b11).addActionListener(new signs());

(b16).addActionListener(new signs());

(b12).addActionListener(new nums());

(b13).addActionListener(new nums());

(b14).addActionListener(new nums());

(b17).addActionListener(new nums());

(b18).addActionListener(new nums());

(b19).addActionListener(new nums());

(b22).addActionListener(new nums());

(b23).addActionListener(new nums());

(b24).addActionListener(new nums());

(b28).addActionListener(new nums());

(b5).addActionListener(new ops());

(b10).addActionListener(new ops());

(b15).addActionListener(new ops());

(b20).addActionListener(new ops());

(b21).addActionListener(new ops());

(b25).addActionListener(new ops());

(b26).addActionListener(new cons());

(b27).addActionListener(new cons());

(b7).addActionListener(new des());

(b8).addActionListener(new des());

(bj1).addActionListener(new tri());

(bj2).addActionListener(new tri());

(bj3).addActionListener(new tri());

(bj4).addActionListener(new tri());

(bj5).addActionListener(new tri());

(bj6).addActionListener(new tri());

b29.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

if(!tf.getText().contains(".")) {

tf.setText(tf.getText()+".");

ta.setText(ta.getText()+".");

}

}

});

b30.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

num2=Double.parseDouble(tf.getText());

if(op!=0) {

count();

}

op=0;

}

});

b1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JDialog jd=new JDialog();

jd.setLayout(new GridLayout(2,4));

jd.add(bj1);

jd.add(bj2);

jd.add(bj3);

jd.add(bj7);

jd.add(bj4);

jd.add(bj5);

jd.add(bj6);

jd.add(bj8);

jd.setTitle("三角学");

jd.setVisible(true);

jd.setBounds(493,358, 300,100);

jd.setResizable(false);

}

});

bj7.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

degree=false;

bj7.setBackground(Color.LIGHT\_GRAY);

bj8.setBackground(Color.white);

}

});

bj8.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

degree=true;

bj8.setBackground(Color.LIGHT\_GRAY);

bj7.setBackground(Color.white);

}

});

tf.setEditable(false);

ta.setEditable(false);

ta.setLineWrap(true);

f.setResizable(false);

f.setVisible(true);

f.setBounds(1000,300, 550,300);

f.setLocationRelativeTo(null);

f.setDefaultCloseOperation(WindowConstants.EXIT\_ON\_CLOSE);

}

class signs implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

if(op==0) {

zero=false;

String str=e.getActionCommand();

if(str=="lg") {

if(Double.parseDouble(tf.getText())>0) {

ta.setText(ta.getText()+"→ lg"+tf.getText());

num1=Math.log10(Double.parseDouble(tf.getText()));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("真数不能小于或者等于零！");

}

}

if(str=="ln") {

if(Double.parseDouble(tf.getText())>0) {

ta.setText(ta.getText()+"→ ln"+tf.getText());

num1=Math.log(Double.parseDouble(tf.getText()));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x不能小于或者等于零！");

}

}

if(str=="x^2") {

ta.setText(ta.getText()+"→ "+tf.getText()+"^2");

num1=Math.pow(Double.parseDouble(tf.getText()), 2);

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="sqrt") {

if(Double.parseDouble(tf.getText())>=0) {

ta.setText(ta.getText()+"→ sqrt"+tf.getText());

num1=Math.pow(Double.parseDouble(tf.getText()), 0.5);

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x不能小于零！");

}

}

if(str=="%") {

ta.setText(ta.getText()+"→ "+tf.getText()+"%");

num1=Double.parseDouble(tf.getText())/100;

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="x!") {

if((int)Double.parseDouble(tf.getText())==Double

.parseDouble(tf.getText())&&Double.parseDouble(tf.getText())>=0) {

num1=1;

ta.setText(ta.getText()+"→ "+tf.getText()+"!");

for(int i=1;i<=(int)Double.parseDouble(tf.getText());i++) {

num1\*=i;

}

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x必须为正整数!");

}

}

if(str=="1/x") {

if(Double.parseDouble(tf.getText())!=0) {

ta.setText(ta.getText()+"→ 1/"+tf.getText());

num1=1/Double.parseDouble(tf.getText());

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x不能为零!");

}

}

result=true;

addnum=true;

}

else {

zero=false;

String str=e.getActionCommand();

if(str=="lg") {

if(Double.parseDouble(tf.getText())>0) {

ta.setText(ta.getText()+"→ "+num1+opstr+"lg"+tf.getText());

num2=Math.log10(Double.parseDouble(tf.getText()));

tf.setText(String.valueOf(num2));

}

else {

tf.setText("真数不能小于或者等于零！");

}

}

if(str=="ln") {

if(Double.parseDouble(tf.getText())>0) {

ta.setText(ta.getText()+"→ "+num1+opstr+"ln"+tf.getText());

num2=Math.log(Double.parseDouble(tf.getText()));

tf.setText(String.valueOf(num2));

}

else {

tf.setText("x不能小于或者等于零！");

}

}

if(str=="x^2") {

ta.setText(ta.getText()+"→ "+num1+opstr+tf.getText()+"^2");

num2=Math.pow(Double.parseDouble(tf.getText()), 2);

tf.setText(String.valueOf(num2));

}

if(str=="sqrt") {

if(Double.parseDouble(tf.getText())>=0) {

ta.setText(ta.getText()+"→ "+num1+opstr+"sqrt"+tf.getText());

num2=Math.pow(Double.parseDouble(tf.getText()), 0.5);

tf.setText(String.valueOf(num2));

}

else {

tf.setText("x不能小于零！");

}

}

if(str=="%") {

ta.setText(ta.getText()+"→ "+num1+opstr+tf.getText()+"%");

num2=Double.parseDouble(tf.getText())/100;

tf.setText(String.valueOf(num2));

}

if(str=="x!") {

if((int)Double.parseDouble(tf.getText())==Double

.parseDouble(tf.getText())&&Double.parseDouble(tf.getText())>=0) {

num2=1;

ta.setText(ta.getText()+"→ "+num1+opstr+tf.getText()+"!");

for(int i=1;i<=(int)Double.parseDouble(tf.getText());i++) {

num2\*=i;

}

tf.setText(String.valueOf(num2));

}

else {

tf.setText("x必须为正整数!");

}

}

if(str=="1/x") {

if(Double.parseDouble(tf.getText())!=0) {

ta.setText(ta.getText()+"→ 1/"+tf.getText());

num2=1/Double.parseDouble(tf.getText());

tf.setText(String.valueOf(num2));

ta.setText(ta.getText()+"="+num2+" ");

}

else {

tf.setText("x不能为零!");

}

}

result=true;

addnum=true;

}

}

}

class nums implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

String str=e.getActionCommand();

if(tf.getText().equals("0")) {

tf.setText(str);

}

else

tf.setText(tf.getText()+str);

if(opflag==true&&numflag2==false) {

tf.setText(str);

numflag2=true;

}

if(addnum&&op==0) {

tf.setText(str);

addnum=false;

num1=Double.parseDouble(tf.getText());

}

if(!tf.getText().equals("0")) {

ta.setText(ta.getText()+tf.getText().charAt(tf.getText().length()-1));

}

else {

if(zero==false)

ta.setText(ta.getText()+0);

zero=true;

}

}

}

class ops implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

zero=false;

String str=e.getActionCommand();

opflag=true;

if (numflag1==false){

num1=Double.parseDouble(tf.getText());

numflag1=true;

}

else {

num2=Double.parseDouble(tf.getText());

count();

numflag2=false;

}

if(str=="+") {

op=1;

opstr="+";

if(!result)

ta.setText(ta.getText()+"+");

else {

ta.setText(ta.getText()+num1+"+");

}

}

if(str=="-") {

op=2;

opstr="-";

if(!result)

ta.setText(ta.getText()+"-");

else {

ta.setText(ta.getText()+num1+"-");

}

}

if(str=="×") {

op=3;

opstr="×";

if(!result)

ta.setText(ta.getText()+"×");

else {

ta.setText(ta.getText()+num1+"×");

}

}

if(str=="÷") {

op=4;

opstr="÷";

if(!result)

ta.setText(ta.getText()+"÷");

else {

ta.setText(ta.getText()+num1+"÷");

}

}

if(str=="mod") {

op=5;

opstr="(mod)";

if(!result)

ta.setText(ta.getText()+"(mod)");

else {

ta.setText(ta.getText()+num1+"(mod)");

}

}

if(str=="x^y") {

op=6;

opstr="^";

if(!result)

ta.setText(ta.getText()+"^");

else {

ta.setText(ta.getText()+num1+"^");

}

}

}

}

class cons implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

String str=e.getActionCommand();

if(tf.getText().equals("0")) {

if(str=="π") {

tf.setText(Math.PI+"");

}

else {

tf.setText(Math.E+"");

}

}

else {

if(str=="π") {

tf.setText(Double.parseDouble(tf.getText())\*Math.PI+"");

}

else {

tf.setText(Double.parseDouble(tf.getText())\*Math.E+"");

}

}

if(opflag==true&&numflag2==false) {

if(str=="π") {

tf.setText(Math.PI+"");

}

else {

tf.setText(Math.E+"");

}

numflag2=true;

}

if(str=="π") {

ta.setText(ta.getText()+"π");

}

else {

ta.setText(ta.getText()+"e");

}

}

}

class des implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

String str=e.getActionCommand();

if(str=="c") {

tf.setText("0");

ta.setText("");

op=0;

numflag1=false;

numflag2=false;

opflag=false;

result=false;

}

else {

tf.setText(tf.getText().substring(0, tf.getText().length()-1));

ta.setText(ta.getText().substring(0, ta.getText().length()-1));

}

}

}

class tri implements ActionListener{

@Override

public void actionPerformed(ActionEvent e) {

zero=false;

String str=e.getActionCommand();

if(!degree) {

if(str=="sinx") {

ta.setText(ta.getText()+"→ sin"+tf.getText()+"°");

num1=Math.sin(Math.toRadians(Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="cosx") {

ta.setText(ta.getText()+"→ cos"+tf.getText()+"°");

num1=Math.cos(Math.toRadians(Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="tanx") {

ta.setText(ta.getText()+"→ tan"+tf.getText()+"°");

num1=Math.tan(Math.toRadians(Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="asinx") {

ta.setText(ta.getText()+"→ arcsin"+tf.getText());

num1=Math.toDegrees(Math.asin((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+"° ");

}

if(str=="acosx") {

ta.setText(ta.getText()+"→ acos"+tf.getText());

num1=Math.toDegrees(Math.acos((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+"° ");

}

if(str=="atanx") {

ta.setText(ta.getText()+"→ arctan"+tf.getText());

num1=Math.toDegrees(Math.atan((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+"° ");

}

}

else {

if(str=="sinx") {

ta.setText(ta.getText()+"→ sin"+tf.getText()+"");

num1=Math.sin((Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="cosx") {

ta.setText(ta.getText()+"→ cos"+tf.getText()+"");

num1=Math.cos((Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="tanx") {

ta.setText(ta.getText()+"→ tan"+tf.getText()+"");

num1=Math.tan((Double.parseDouble(tf.getText())));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

if(str=="asinx") {

if(Math.abs(Double.parseDouble(tf.getText()))<=1) {

ta.setText(ta.getText()+"→ arcsin"+tf.getText());

num1=(Math.asin((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x的绝对值不能大于1!");

}

}

if(str=="acosx") {

if(Math.abs(Double.parseDouble(tf.getText()))<=1) {

ta.setText(ta.getText()+"→ acos"+tf.getText());

num1=(Math.acos((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

else {

tf.setText("x的绝对值不能大于1!");

}

}

if(str=="atanx") {

ta.setText(ta.getText()+"→ arctan"+tf.getText());

num1=(Math.atan((Double.parseDouble(tf.getText()))));

tf.setText(String.valueOf(num1));

ta.setText(ta.getText()+"="+num1+" ");

}

}

result=true;

addnum=true;

}

}

public void count () {

switch (op) {

case 1:

num1=num1+num2;

tf.setText(num1+"");

break;

case 2:

num1=num1-num2;

tf.setText(num1+"");

break;

case 3:

num1=num1\*num2;

tf.setText(num1+"");

break;

case 4:

num1=num1/num2;

tf.setText(num1+"");

break;

case 5:

if((int)num1==num1&&(int)num2==num2){

num1=num1%num2;

tf.setText(num1+"");

}

else {

tf.setText("两个整数才可以求余！");

}

break;

case 6:

num1=Math.pow(num1, num2);

tf.setText(num1+"");

break;

}

ta.setText(ta.getText().substring(0, ta.getText().length())+"="+num1+" → ");

result=true;

addnum=true;

op=0;

}

public static void main(String[] args) {

new Calculator();

}

}