package com.learnjava;  
  
import java.awt.BorderLayout;  
import java.awt.Button;  
import java.awt.Font;  
import java.awt.Frame;  
import java.awt.GridLayout;  
import java.awt.Panel;  
import java.awt.TextField;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
import java.awt.event.WindowAdapter;  
import java.awt.event.WindowEvent;  
import java.awt.event.WindowListener;  
class Numpan extends Panel implements ActionListener{  
 Button n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,point,equal;  
 Button plus,minus,multiply, divide;  
 Button m\_plus,m\_minus,clear,advanced;  
 TextField display;  
 String op1,op2,result;  
 String op\_flag;  
 String data;  
 double dop1,dop2,dresult;  
 boolean flag\_advanced=true;  
 public Numpan(TextField display) {  
 this.display = display;  
 setLayout(new GridLayout(0,4));  
 n0 = new Button("0");  
 n0.setActionCommand("zero");  
 n0.addActionListener(this);  
 n1 = new Button("1");  
 n1.setActionCommand("one");  
 n1.addActionListener(this);  
 n2 = new Button("2");  
 n2.setActionCommand("two");  
 n2.addActionListener(this);  
 n3 = new Button("3");  
 n3.setActionCommand("three");  
 n3.addActionListener(this);  
 n4 = new Button("4");  
 n4.setActionCommand("four");  
 n4.addActionListener(this);  
 n5 = new Button("5");  
 n5.setActionCommand("five");  
 n5.addActionListener(this);  
 n6 = new Button("6");  
 n6.setActionCommand("six");  
 n6.addActionListener(this);  
 n7 = new Button("7");  
 n7.setActionCommand("seven");  
 n7.addActionListener(this);  
 n8 = new Button("8");  
 n8.setActionCommand("eight");  
 n8.addActionListener(this);  
 n9 = new Button("9");  
 n9.setActionCommand("nine");  
 n9.addActionListener(this);  
 point = new Button(".");  
 point.setActionCommand("point");  
 point.addActionListener(this);  
 equal = new Button("=");  
 equal.setActionCommand("equal");  
 equal.addActionListener(this);  
 plus = new Button("+");  
 plus.setActionCommand("plus");  
 plus.addActionListener(this);  
 minus = new Button("-");  
 minus.setActionCommand("minus");  
 minus.addActionListener(this);  
 multiply = new Button("x");  
 multiply.setActionCommand("multiply");  
 multiply.addActionListener(this);  
 divide = new Button("/");  
 divide.setActionCommand("divide");  
 divide.addActionListener(this);  
 m\_plus = new Button("M+");  
 m\_plus.setActionCommand("m\_plus");  
 m\_plus.addActionListener(this);  
 m\_minus = new Button("M-");  
 m\_minus.setActionCommand("m\_minus");  
 m\_minus.addActionListener(this);  
 clear = new Button("C");  
 clear.setActionCommand("clear");  
 clear.addActionListener(this);  
 advanced = new Button("ADV");  
 advanced.setActionCommand("advanced");  
 advanced.addActionListener(this);  
 add(m\_plus);  
 add(m\_minus);  
 add(clear);  
 add(advanced);  
 add(n1);  
 add(n2);  
 add(n3);  
 add(plus);  
 add(n4);  
 add(n5);  
 add(n6);  
 add(minus);  
 add(n7);  
 add(n8);  
 add(n9);  
 add(multiply);  
 add(point);  
 add(n0);  
 add(equal);  
 add(divide);  
 }  
 public String getDisplayText(){  
 return display.getText().toString();  
 }  
 public void setDisplay(String text){  
 display.setText(text);  
 }  
 public void clearDisplay(){  
 System.*out*.println("Clear Called");  
 setDisplay("");  
 data = "";  
 }  
 public void changeAdvanced(boolean toAdvanced){  
 if(toAdvanced){  
 plus.setLabel("sin");  
 plus.setActionCommand("sin");  
//System.out.println("cos in");  
 minus.setLabel("cos");  
 minus.setActionCommand("cos");  
//System.out.println("cos out");  
 multiply.setLabel("tan");  
 multiply.setActionCommand("tan");  
 divide.setLabel("log");  
 divide.setActionCommand("log");  
 }  
 else{  
 plus.setLabel("+");  
 plus.setActionCommand("plus");  
 minus.setLabel("-");  
 minus.setActionCommand("minus");  
 multiply.setLabel("x");  
 multiply.setActionCommand("multiply");  
 divide.setLabel("/");  
 divide.setActionCommand("divide");  
 }  
 }  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 data = getDisplayText();  
 switch(e.getActionCommand()){  
 case "zero":  
 setDisplay(data+"0");  
 break;  
 case "one":  
 setDisplay(data+"1");  
 break;  
 case "two":  
 setDisplay(data+"2");  
 break;  
 case "three":  
 setDisplay(data+"3");  
 break;  
 case "four":  
 setDisplay(data+"4");  
 break;  
 case "five":  
 setDisplay(data+"5");  
 break;  
 case "six":  
 setDisplay(data+"6");  
 break;  
 case "seven":  
 setDisplay(data+"7");  
 break;  
 case "eight":  
 setDisplay(data+"8");  
 break;  
 case "nine":  
 setDisplay(data+"9");  
 break;  
 case "plus":  
 op1 = data;  
 op\_flag = "plus";  
 clearDisplay();  
 break;  
 case "minus":  
 op1 = data;  
 op\_flag = "minus";  
 clearDisplay();  
 break;  
 case "multiply":  
 op1 = data;  
 op\_flag = "multiply";  
 clearDisplay();  
 break;  
 case "divide":  
 op1 = data;  
 op\_flag = "divide";  
 clearDisplay();  
 break;  
 case "clear":  
 clearDisplay();  
 break;  
 case "advanced":  
 if(flag\_advanced){  
 changeAdvanced(true);  
 flag\_advanced = false;  
 }  
 else{  
 changeAdvanced(false);  
 flag\_advanced = true;  
 }  
 break;  
 case "sin":  
 op1 = data;  
 setDisplay(String.*valueOf*(Math.*sin*(Double.*valueOf*(op1))));  
 break;  
 case "cos":  
 op1 = data;  
 setDisplay(String.*valueOf*(Math.*cos*(Double.*valueOf*(op1))));  
 break;  
 case "tan":  
 op1 = data;  
 setDisplay(String.*valueOf*(Math.*tan*(Double.*valueOf*(op1))));  
 break;  
 case "log":  
 op1 = data;  
 setDisplay(String.*valueOf*(Math.*log*(Double.*valueOf*(op1))));  
 break;  
 case "equal":  
 switch(op\_flag){  
 case "plus":  
 op2 = data;  
 clearDisplay();  
 dop1 = Double.*parseDouble*(op1);  
 dop2 = Double.*parseDouble*(op2);  
 dresult = dop1 + dop2;  
 result = String.*valueOf*(dresult);  
 setDisplay(result);  
 op\_flag = "";  
 break;  
 case "minus":  
 op2 = data;  
 clearDisplay();  
 dop1 = Double.*parseDouble*(op1);  
 dop2 = Double.*parseDouble*(op2);  
 dresult = dop1 - dop2;  
 result = String.*valueOf*(dresult);  
 setDisplay(result);  
 op\_flag = "";  
 break;  
 case "multiply":  
 op2 = data;  
 clearDisplay();  
 dop1 = Double.*parseDouble*(op1);  
 dop2 = Double.*parseDouble*(op2);  
 dresult = dop1 \* dop2;  
 result = String.*valueOf*(dresult);  
 setDisplay(result);  
 op\_flag = "";  
 break;  
 case "divide":  
 op2 = data;  
 clearDisplay();  
 dop1 = Double.*parseDouble*(op1);  
 dop2 = Double.*parseDouble*(op2);  
 dresult = dop1 / dop2;  
 result = String.*valueOf*(dresult);  
 setDisplay(result);  
 op\_flag = "";  
 break;  
 }  
 }  
 }  
}  
class Calculator extends Frame {  
 TextField display;  
 public Calculator() {  
 display = new TextField();  
 display.setFont(new Font("Times New Roman", Font.*BOLD*, 50));  
 setLayout(new BorderLayout());  
 add(new Numpan(display),BorderLayout.*CENTER*);  
 add(display,BorderLayout.*NORTH*);  
 setVisible(true);  
 setSize(500,500);  
 addWindowListener(new WindowAdapter() {  
 @Override  
 public void windowClosing(WindowEvent e) {  
 dispose();  
 }  
 });  
 }  
}  
public class Main {  
 public static void main(String[] args) {  
 new Calculator();  
 }  
}