CALCULATOR

package Swings;

import java.lang.\*;

import java.util.Scanner.\*;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class Calculator extends JFrame implements ActionListener, KeyListener

{

JFrame mainframe = new JFrame("Calculator");

JButton zero,one,two,three,four,five,six,seven,eight,nine,point,plus,minus,star,divide,sqroot,square,clear,equal;

JTextField question,answer;

JLabel qlabel,alabel;

double a=0,b=0,res=0;

int op=0;

Calculator()

{

mainframe.setLayout(null);;

mainframe.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

mainframe.setResizable(false);

Dimension dim = Toolkit.getDefaultToolkit().getScreenSize();

mainframe.setLocation((dim.width-310)/2, (dim.height-350)/2);

qlabel = new JLabel("PROBLEM");

qlabel.setBounds(10,10,100,30);

qlabel.setBackground(Color.gray);

mainframe.add(qlabel);

alabel = new JLabel("RESULT");

alabel.setBounds(10,50,100,30);

alabel.setBackground(Color.gray);

mainframe.add(alabel);

question = new JTextField();

mainframe.addKeyListener(this);

question.setBounds(110,10,150,30);

question.setBackground(Color.WHITE);

mainframe.add(question);

answer = new JTextField();

answer.setBounds(110,50,150,30);

answer.setBackground(Color.WHITE);

mainframe.add(answer);

seven = new JButton("7");

seven.setBounds(25,100,45,45);

seven.setBackground(Color.LIGHT\_GRAY);

mainframe.add(seven);

eight = new JButton("8");

eight.setBounds(75,100,45,45);

eight.setBackground(Color.LIGHT\_GRAY);

mainframe.add(eight);

nine = new JButton("9");

nine.setBounds(125,100,45,45);

nine.setBackground(Color.LIGHT\_GRAY);

mainframe.add(nine);

plus = new JButton("+");

plus.setBounds(175,100,45,45);

plus.setBackground(Color.LIGHT\_GRAY);

mainframe.add(plus);

minus = new JButton("-");

minus.setBounds(225,100,45,45);

minus.setBackground(Color.LIGHT\_GRAY);

mainframe.add(minus);

four = new JButton("4");

four.setBounds(25,150,45,45);

four.setBackground(Color.LIGHT\_GRAY);

mainframe.add(four);

five = new JButton("5");

five.setBounds(75,150,45,45);

five.setBackground(Color.LIGHT\_GRAY);

mainframe.add(five);

six = new JButton("6");

six.setBounds(125,150,45,45);

six.setBackground(Color.LIGHT\_GRAY);

mainframe.add(six);

star = new JButton("\*");

star.setBounds(175,150,45,45);

star.setBackground(Color.LIGHT\_GRAY);

mainframe.add(star);

divide = new JButton("/");

divide.setBounds(225,150,45,45);

divide.setBackground(Color.LIGHT\_GRAY);

mainframe.add(divide);

one = new JButton("1");

one.setBounds(25,200,45,45);

one.setBackground(Color.LIGHT\_GRAY);

mainframe.add(one);

two = new JButton("2");

two.setBounds(75,200,45,45);

two.setBackground(Color.LIGHT\_GRAY);

mainframe.add(two);

three = new JButton("3");

three.setBounds(125,200,45,45);

three.setBackground(Color.LIGHT\_GRAY);

mainframe.add(three);

sqroot = new JButton("/x");

sqroot.setBounds(175,200,45,45);

sqroot.setBackground(Color.LIGHT\_GRAY);

mainframe.add(sqroot);

square = new JButton("S");

square.setBounds(225,200,45,45);

square.setBackground(Color.LIGHT\_GRAY);

mainframe.add(square);

point= new JButton(".");

point.setBounds(25,250,45,45);

point.setBackground(Color.LIGHT\_GRAY);

mainframe.add(point);

zero = new JButton("0");

zero.setBounds(75,250,45,45);

zero.setBackground(Color.LIGHT\_GRAY);

mainframe.add(zero);

equal = new JButton("=");

equal.setBounds(125,250,70,45);

equal.setBackground(Color.LIGHT\_GRAY);

mainframe.add(equal);

clear = new JButton("C");

clear.setBounds(200,250,70,45);

clear.setBackground(Color.LIGHT\_GRAY);

mainframe.add(clear);

zero.addActionListener(this);

one.addActionListener(this);

two.addActionListener(this);

three.addActionListener(this);

four.addActionListener(this);

five.addActionListener(this);

six.addActionListener(this);

seven.addActionListener(this);

eight.addActionListener(this);

nine.addActionListener(this);

plus.addActionListener(this);

minus.addActionListener(this);

star.addActionListener(this);

divide.addActionListener(this);

sqroot.addActionListener(this);

square.addActionListener(this);

equal.addActionListener(this);

clear.addActionListener(this);

point.addActionListener(this);

mainframe.setSize(310,350);

mainframe.setVisible(true);

}

public static void main(String[] args)

{

// TODO Auto-generated method stub

new Calculator();

}

public void actionPerformed(ActionEvent e)

{

if(e.getSource()==zero)

question.setText(question.getText().concat("0"));

if(e.getSource()==one)

question.setText(question.getText().concat("1"));

if(e.getSource()==two)

question.setText(question.getText().concat("2"));

if(e.getSource()==three)

question.setText(question.getText().concat("3"));

if(e.getSource()==four)

question.setText(question.getText().concat("4"));

if(e.getSource()==five)

question.setText(question.getText().concat("5"));

if(e.getSource()==six)

question.setText(question.getText().concat("6"));

if(e.getSource()==seven)

question.setText(question.getText().concat("7"));

if(e.getSource()==eight)

question.setText(question.getText().concat("8"));

if(e.getSource()==nine)

question.setText(question.getText().concat("9"));

if(e.getSource()==point)

question.setText(question.getText().concat("."));

if(e.getSource()==plus)

{

a= Double.parseDouble(question.getText());

op=1;

question.setText("");

}

if(e.getSource()==minus)

{

a= Double.parseDouble(question.getText());

op=2;

question.setText("");

}

if(e.getSource()==star)

{

a= Double.parseDouble(question.getText());

op=3;

question.setText("");

}

if(e.getSource()==divide)

{

a= Double.parseDouble(question.getText());

op=4;

question.setText("");

}

if(e.getSource()==sqroot)

{

a= Double.parseDouble(question.getText());

op=5;

}

if(e.getSource()==square)

{

a= Double.parseDouble(question.getText());

op=6;

}

if(e.getSource()==equal)

{

b = Double.parseDouble(question.getText());

switch(op)

{

case 1 : res = a+b;

break;

case 2 : res = a-b;

break;

case 3 : res = a\*b;

break;

case 4 : res = a/b;

break;

case 5 : res = Math.sqrt(a);

break;

case 6 : res = a\*a;

break;

default : res=0;

}

answer.setText(""+res);

}

if(e.getSource()==clear)

{

question.setText("");

answer.setText("");

}

}

@Override

public void keyPressed(KeyEvent e) {

// TODO Auto-generated method stub

int key = e.getKeyCode();

if(key=='0')

question.setText(question.getText().concat("0"));

if(key=='1')

question.setText(question.getText().concat("1"));

if(key=='2')

question.setText(question.getText().concat("2"));

if(key=='3')

question.setText(question.getText().concat("3"));

if(key=='4')

question.setText(question.getText().concat("4"));

if(key=='5')

question.setText(question.getText().concat("5"));

if(key=='6')

question.setText(question.getText().concat("6"));

if(key=='7')

question.setText(question.getText().concat("7"));

if(key=='8')

question.setText(question.getText().concat("8"));

if(key=='9')

question.setText(question.getText().concat("9"));

if(key=='+')

{

a= Double.parseDouble(question.getText());

op=1;

question.setText("");

}

if(key=='-')

{

a= Double.parseDouble(question.getText());

op=1;

question.setText("");

}

if(key=='\*')

{

a= Double.parseDouble(question.getText());

op=1;

question.setText("");

}

if(key=='/')

{

a= Double.parseDouble(question.getText());

op=1;

question.setText("");

}

}

@Override

public void keyReleased(KeyEvent arg0) {

// TODO Auto-generated method stub

}

@Override

public void keyTyped(KeyEvent arg0) {

// TODO Auto-generated method stub

}

}

EDITOR

package Swings;

import java.awt.\*;

import java.awt.event.\*;

import java.awt.event.KeyEvent;

import java.awt.image.ImageObserver;

import java.awt.image.ImageProducer;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Scanner;

import java.io.\*;

import java.io.File;

import java.io.PrintWriter;

import java.io.BufferedWriter;

import java.io.FileReader;

import java.io.FileWriter;

import javax.swing.\*;

import javax.swing.\*;

public class Notepad extends JFrame implements ActionListener{

JFrame mainframe= new JFrame();

JFrame vhelp, about;

JTextArea textarea = new JTextArea();

JFileChooser filechooser;

JMenuBar menubar= new JMenuBar();

JMenu file,edit,help;

JMenuItem newf,open,save,saveas,quit,cut,copy,paste,delete,selectall,viewhelp,aboutnp;

Notepad()

{

mainframe.setSize(1000,700);

mainframe.setTitle("NOTEPAD");

filechooser = new JFileChooser();

file = new JMenu("File");

edit = new JMenu("Edit");

help = new JMenu("Help");

newf = new JMenuItem("New");

open = new JMenuItem("Open");

save = new JMenuItem("Save");

saveas = new JMenuItem("Save As");

quit = new JMenuItem("Quit");

cut = new JMenuItem("Cut Ctrl+X");

copy = new JMenuItem("Copy Ctrl+C");

paste = new JMenuItem("Paste Ctrl+V");

delete = new JMenuItem("Delete");

selectall = new JMenuItem("Select All Ctrl+A");

viewhelp = new JMenuItem("View Help");

aboutnp = new JMenuItem("About Notepad");

newf.addActionListener(this);

OpenListener openL = new OpenListener();

open.addActionListener(openL);

SaveListener saveL = new SaveListener();

save.addActionListener(saveL);

saveas.addActionListener(saveL);

quit.addActionListener(this);

cut.addActionListener(this);

copy.addActionListener(this);

paste.addActionListener(this);

delete.addActionListener(this);

selectall.addActionListener(this);

viewhelp.addActionListener(this);

aboutnp.addActionListener(this);

file.add(newf);

file.add(open);

file.add(save);

file.add(saveas);

file.add(quit);

edit.add(cut);

edit.add(copy);

edit.add(paste);

edit.add(delete);

edit.add(selectall);

help.add(viewhelp);

help.add(aboutnp);

mainframe.setLayout(null);

menubar.setBounds(0,0,1000,30);

menubar.add(file);

menubar.add(edit);

menubar.add(help);

mainframe.add(menubar);

//textarea.setBounds(0,30,1000,700);

JScrollPane scrollpane = new JScrollPane(textarea);

scrollpane.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS);

scrollpane.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_ALWAYS);

scrollpane.setBounds(0,30,mainframe.getWidth()-10,mainframe.getHeight()-60);

mainframe.add(scrollpane);

mainframe.setResizable(false);

mainframe.setIconImage(new ImageIcon("C:\\Users\\USER\\Desktop\\IMG\_20150801\_152938656.jpg").getImage());

mainframe.setVisible(true);

mainframe.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

}

public static void main(String[] args) {

// TODO Auto-generated method stub

new Notepad();

}

@Override

public void actionPerformed(ActionEvent e) {

// TODO Auto-generated method stub

if(e.getSource()==newf){

Thread thread=new Thread(new Runnable() {

@Override

public void run() {

// TODO Auto-generated method stub

new Notepad();

}

});

thread.start();

}

if(e.getSource()==quit)

mainframe.dispose();

if(e.getSource()==cut)

textarea.cut();

if(e.getSource()==copy)

textarea.copy();

if(e.getSource()==paste)

textarea.paste();

if(e.getSource()==delete){

textarea.replaceSelection("");

}

if(e.getSource()==selectall){

textarea.selectAll();

}

if(e.getSource()==viewhelp)

{

vhelp = new JFrame("HELP");

vhelp.setSize(750,650);

JLabel l1 = new JLabel("INSTRUCTIONS TO USE NOTEPAD :");

l1.setBounds(10,10,390,20);

vhelp.add(l1);

JLabel l2 = new JLabel(" NOTEPAD is used to create text files.");

l2.setBounds(10,40,490,20);

vhelp.add(l2);

JLabel l3 = new JLabel("To know about the options present, go to top left corner of the Dialog box.");

l3.setBounds(10,70,490,20);

vhelp.add(l3);

JLabel l4 = new JLabel("When 'File' is selected,");

l4.setBounds(10,100,390,20);

vhelp.add(l4);

JLabel l5 = new JLabel("NEW : To create new file.");

l5.setBounds(10,130,390,20);

vhelp.add(l5);

JLabel l6 = new JLabel("OPEN : To open already existing file.");

l6.setBounds(10,160,390,20);

vhelp.add(l6);

JLabel l7 = new JLabel("SAVE : To save the created text file.");

l7.setBounds(10,190,390,20);

vhelp.add(l7);

JLabel l8 = new JLabel("SAVE AS : To change the name of the existing saved file or to create duplicate copy of file.");

l8.setBounds(10,220,590,20);

vhelp.add(l8);

JLabel l9 = new JLabel("QUIT : To close Notepad");

l9.setBounds(10,250,390,20);

vhelp.add(l9);

JLabel l10 = new JLabel("When 'EDIT' is selected,");

l10.setBounds(10,280,390,20);

vhelp.add(l10);

JLabel l11 = new JLabel("CUT : To cut the selected text");

l11.setBounds(10,310,390,20);

vhelp.add(l11);

JLabel l12 = new JLabel("COPY : To copy the selected text");

l12.setBounds(10,340,390,20);

vhelp.add(l12);

JLabel l13 = new JLabel("PASTE : To paste the cut or copied text");

l13.setBounds(10,370,400,20);

vhelp.add(l13);

JLabel l14 = new JLabel("DELETE : To delete the selected text");

l14.setBounds(10,400,390,20);

vhelp.add(l14);

JLabel l15 = new JLabel("SELECT ALL : To select the entire text.");

l15.setBounds(10,430,390,20);

vhelp.add(l15);

JLabel l16 = new JLabel("");

l16.setBounds(10,460,390,20);

vhelp.add(l16);

vhelp.setResizable(false);

vhelp.setVisible(true);

}

if(e.getSource()==aboutnp)

{

about = new JFrame("ABOUT NOTEPAD");

}

}

class OpenListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

if(JFileChooser.APPROVE\_OPTION==filechooser.showOpenDialog(mainframe))

{

File file = filechooser.getSelectedFile();

textarea.setText("");

Scanner in = null;

try {

in = new Scanner(file);

while(in.hasNext())

{

String line = in.nextLine();

textarea.append(line+"\n");

}

}

catch(Exception ex)

{

ex.printStackTrace();

}

finally

{

in.close();

}

}

}

}

class SaveListener implements ActionListener

{

public void actionPerformed(ActionEvent e)

{

if(JFileChooser.APPROVE\_OPTION==filechooser.showSaveDialog(mainframe))

{

File file = filechooser.getSelectedFile();

PrintWriter out = null;

try {

out = new PrintWriter(file);

String output = textarea.getText();

System.out.println(output);

out.println(output);

}

catch(Exception ex)

{

ex.printStackTrace();

}

finally

{

try

{

out.flush();

}

catch(Exception ex1) {}

try

{

out.close();

}

catch(Exception ex1) {}

}

}

}

}

}