// Java Program to create a text editor using java

import java.awt.\*;

import javax.swing.\*;

import java.io.\*;

import java.awt.event.\*;

import javax.swing.plaf.metal.\*;

import javax.swing.text.\*;

class editor extends JFrame implements ActionListener {

// Text component

JTextArea t;

// Frame

JFrame f;

// Constructor

editor()

{

// Create a frame

f = new JFrame("editor");

try {

// Set metal look and feel

UIManager.setLookAndFeel("javax.swing.plaf.metal.MetalLookAndFeel");

// Set theme to ocean

MetalLookAndFeel.setCurrentTheme(new OceanTheme());

}

catch (Exception e) {

}

// Text component

t = new JTextArea();

// Create a menubar

JMenuBar mb = new JMenuBar();

// Create amenu for menu

JMenu m1 = new JMenu("File");

// Create menu items

JMenuItem mi1 = new JMenuItem("New");

JMenuItem mi2 = new JMenuItem("Open");

JMenuItem mi3 = new JMenuItem("Save");

JMenuItem mi9 = new JMenuItem("Print");

// Add action listener

mi1.addActionListener(this);

mi2.addActionListener(this);

mi3.addActionListener(this);

mi9.addActionListener(this);

m1.add(mi1);

m1.add(mi2);

m1.add(mi3);

m1.add(mi9);

// Create amenu for menu

JMenu m2 = new JMenu("Edit");

// Create menu items

JMenuItem mi4 = new JMenuItem("cut");

JMenuItem mi5 = new JMenuItem("copy");

JMenuItem mi6 = new JMenuItem("paste");

// Add action listener

mi4.addActionListener(this);

mi5.addActionListener(this);

mi6.addActionListener(this);

m2.add(mi4);

m2.add(mi5);

m2.add(mi6);

JMenuItem mc = new JMenuItem("close");

mc.addActionListener(this);

mb.add(m1);

mb.add(m2);

mb.add(mc);

f.setJMenuBar(mb);

f.add(t);

f.setSize(500, 500);

f.show();

}

// If a button is pressed

public void actionPerformed(ActionEvent e)

{

String s = e.getActionCommand();

if (s.equals("cut")) {

t.cut();

}

else if (s.equals("copy")) {

t.copy();

}

else if (s.equals("paste")) {

t.paste();

}

else if (s.equals("Save")) {

// Create an object of JFileChooser class

JFileChooser j = new JFileChooser("f:");

// Invoke the showsSaveDialog function to show the save dialog

int r = j.showSaveDialog(null);

if (r == JFileChooser.APPROVE\_OPTION) {

// Set the label to the path of the selected directory

File fi = new File(j.getSelectedFile().getAbsolutePath());

try {

// Create a file writer

FileWriter wr = new FileWriter(fi, false);

// Create buffered writer to write

BufferedWriter w = new BufferedWriter(wr);

// Write

w.write(t.getText());

w.flush();

w.close();

}

catch (Exception evt) {

JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

// If the user cancelled the operation

else

JOptionPane.showMessageDialog(f, "the user cancelled the operation");

}

else if (s.equals("Print")) {

try {

// print the file

t.print();

}

catch (Exception evt) {

JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

else if (s.equals("Open")) {

// Create an object of JFileChooser class

JFileChooser j = new JFileChooser("f:");

// Invoke the showsOpenDialog function to show the save dialog

int r = j.showOpenDialog(null);

// If the user selects a file

if (r == JFileChooser.APPROVE\_OPTION) {

// Set the label to the path of the selected directory

File fi = new File(j.getSelectedFile().getAbsolutePath());

try {

// String

String s1 = "", sl = "";

// File reader

FileReader fr = new FileReader(fi);

// Buffered reader

BufferedReader br = new BufferedReader(fr);

// Initialize sl

sl = br.readLine();

// Take the input from the file

while ((s1 = br.readLine()) != null) {

sl = sl + "\n" + s1;

}

// Set the text

t.setText(sl);

}

catch (Exception evt) {

JOptionPane.showMessageDialog(f, evt.getMessage());

}

}

// If the user cancelled the operation

else

JOptionPane.showMessageDialog(f, "the user cancelled the operation");

}

else if (s.equals("New")) {

t.setText("");

}

else if (s.equals("close")) {

f.setVisible(false);

}

}

// Main class

public static void main(String args[])

{

editor e = new editor();

}

}



