public class FileCopy {

/\*\*

\* @param args

\*/

public static void main(String[] args) {

FileCopyFrame frame=new FileCopyFrame();

frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT\_ON\_CLOSE);

frame.setSize(500,100);

frame.setVisible(true);

// TODO Auto-generated method stub

}

}

import javax.swing.JFrame;

import javax.swing.JTextField;

import javax.swing.JButton;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.awt.BorderLayout;

import java.awt.GridLayout;

import javax.swing.JPanel;

import javax.swing.JLabel;

import javax.swing.JFileChooser;

public class FileCopyFrame extends JFrame{

JTextField from;

JTextField to;

JButton copy;

JPanel panel;

JButton fromButton;

JButton toButton;

JFileChooser fileChooser1;

JFileChooser fileChooser2;

public FileCopyFrame()

{

setName("File Transfer");

setLayout(new java.awt.BorderLayout());

panel=new JPanel();

from=new JTextField("from");

to=new JTextField("to");

copy=new JButton("copy");

fromButton=new JButton("chooseFile");

toButton=new JButton("chooseDestination");

panel.setLayout(new GridLayout(3,2));

panel.add(new JLabel("from"));

panel.add(from);

panel.add(fromButton);

panel.add(new JLabel("to"));

panel.add(to);

panel.add(toButton);

add(panel,BorderLayout.NORTH);

add(copy,BorderLayout.SOUTH);

fromButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ev)

{

openFromFile();

}

});

toButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ev)

{

openFromFile();

}

});

copy.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent ev)

{

try

{

//System.out.println("inide if");

FileInputStream fin=new FileInputStream(fileChooser1.getSelectedFile());

File file=fileChooser1.getSelectedFile();

int contentLength=(int)file.length();

int offset=0;

byte[] data = new byte[contentLength];

int bytesRead = 0;

while (offset < contentLength) {

bytesRead = fin.read(data, offset, data.length-offset);

if (bytesRead == -1) break;

offset += bytesRead;

}

fin.close();

FileOutputStream fout=new FileOutputStream(to.getText()+file.getName());

fout.write(data);

fout.close();

}

catch(Exception ex)

{

ex.printStackTrace();

}

}

});

}

public void openFromFile()

{

fileChooser1=new JFileChooser();

fileChooser1.setFileSelectionMode(JFileChooser.FILES\_ONLY);

int result=fileChooser1.showOpenDialog(this);

if(result==JFileChooser.CANCEL\_OPTION)

openFromFile();

from.setText(fileChooser1.getSelectedFile().getAbsolutePath());

}

public void openToFile()

{

fileChooser2=new JFileChooser();

fileChooser2.setFileSelectionMode(JFileChooser.DIRECTORIES\_ONLY);

int result=fileChooser2.showOpenDialog(this);

if(result==JFileChooser.CANCEL\_OPTION)

openToFile();

to.setText(fileChooser2.getSelectedFile().getAbsolutePath());

}

}