**Code for secrecy of text msg using steganography**

import java.awt.\*;

import javax.swing.JFrame;

import javax.swing.JButton;

import java.awt.AWTEvent;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JLabel;

import javax.swing.ImageIcon;

import java.io.\*;

import javax.imageio.\*;

import java.awt.image.\*;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

import javax.swing.JScrollPane;

import javax.swing.JSplitPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import javax.swing.border.LineBorder;

import javax.swing.border.\*;

public class homescreen implements ActionListener

{

public static String path;

JButton b1=new JButton("IMAGE STEGANOGRAPHY");

JButton b2=new JButton("AUDIO STEGANOGRAPHY");

JButton b3=new JButton("Help");

JButton b4=new JButton("EXIT");

JFrame f=new JFrame(" SECRECY OF TEXT MESSAGE USING STEGANOGRAPHY");

void home()

{

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\homescreen.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(900,800,1367,720);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

f.add(lable);

f.setExtendedState(Frame.MAXIMIZED\_BOTH);

f.setUndecorated(false);

f.setVisible(true);

b1.setBounds(242,550,290,50);

b1.setBackground(Color.WHITE);

b2.setBounds(821,550,290,50);

b3.setBounds(1270,660,60,40);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

b4.setBounds(530,650,290,50);

b4.setBackground(Color.WHITE);

Font font = new Font("Calisto MT", Font.BOLD,20);

b1.setFont(font);

b2.setFont(font);

b4.setFont(font);

lable.add(b2);

lable.add(b3);

lable.add(b1);

lable.add(b4);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

String str = e.getActionCommand();

if(str.equals("EXIT"))

{

f.dispose();

}

if(str.equals("Help"))

{

homescreen hr= new homescreen();

hr.helppage();

}

if(str.equals("AUDIO STEGANOGRAPHY"))

{

home1 h=new home1();

}

if(str.equals("IMAGE STEGANOGRAPHY"))

{

imgx l=new imgx();

}

}

void helppage()

{

String path;

JFrame fr=new JFrame(" HELP");

fr.setSize(750,489);

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\helpimage.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(900,800,745,460);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

fr.add(lable);

fr.setUndecorated(false);

fr.setLocationRelativeTo(null);

fr.setVisible(true);

fr.setResizable(false);

}

public static void main(String s[])

{

homescreen h =new homescreen();

h.home();

}

}

class imgx extends JFrame implements ActionListener

{

JFrame f =new JFrame(" IMAGE STEGANOGRAPHY");

JButton b1=new JButton("EMBED");

JButton b2=new JButton("EXTRACT");

JButton b3=new JButton(" BACK ");

String path;

public imgx()

{

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\image.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(900,800,1367,720);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

b1.setBounds(242,550,290,50);

b2.setBounds(821,550,290,50);

b3.setBounds(530,650,290,50);

b1.setBackground(Color.WHITE);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

Font font = new Font("Calisto MT", Font.BOLD,20);

b1.setFont(font);

b2.setFont(font);

b3.setFont(font);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

f.setExtendedState(Frame.MAXIMIZED\_BOTH);

f.add(b1);

f.add(b2);

f.add(b3);

f.add(lable);

f.setUndecorated(false);

f.setLocationRelativeTo(null);

f.setVisible(true);

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if(s.equals("EMBED"))

{

EmbedMessage x =new EmbedMessage();

}

else if(s.equals("EXTRACT"))

{

DecodeMessage1 y=new DecodeMessage1();

}

else if(s.equals(" BACK "))

{

f.dispose();

}

}

}

class EmbedMessage extends JFrame implements ActionListener

{

int a=0;

JLabel l1=new JLabel(" IMAGE FILE :");

JTextArea tf1=new JTextArea(10,3);

JLabel l2=new JLabel("TEXT MESSAGE :");

JLabel l3=new JLabel(" WATERMARK :");

JLabel l4=new JLabel(" OUTPUT FILE :");

JTextField tf2=new JTextField(35);

JButton b1=new JButton("OK");

JButton b2=new JButton("CANCEL");

JButton b3=new JButton("BROWSE");

JButton b4=new JButton("EMBED");

JButton b5=new JButton(" BROWSE");

JButton reset = new JButton("RESET");

JTextArea message = new JTextArea(10,3);

JTextArea message1 = new JTextArea(10,3);

JTextArea message2 = new JTextArea(10,3);

BufferedImage sourceImage = null, embeddedImage = null;

JSplitPane sp = new JSplitPane(JSplitPane.HORIZONTAL\_SPLIT);

JScrollPane originalPane = new JScrollPane(),

embeddedPane = new JScrollPane();

String path;

JFrame f=new JFrame();

public EmbedMessage()

{

super("Embed stegonographic message in image");

Font font = new Font("Calisto MT", Font.BOLD,20);

Font font1 = new Font("Calisto MT", Font.BOLD,13);

l1.setFont(font);

l2.setFont(font);

l3.setFont(font);

l4.setFont(font);

l1.setForeground(Color.WHITE);

l1.setForeground(Color.WHITE);

l3.setForeground(Color.WHITE);

l4.setForeground(Color.WHITE);

l2.setForeground(Color.WHITE);

l3.setFont(font);

l1.setBounds(112,110,198,40);

tf1.setBounds(315,120,130,25);

b3.setBounds(470,114,95,30);

b5.setBounds(470,396,100,30);

l2.setBounds(112,160,218,40);

l3.setBounds(112,324,218,40);

l4.setBounds(112,390,218,40);

message2.setBounds(317,400,130,25);

message.setBounds(313,166,250,150);

message1.setBounds(315,330,250,40);

b1.setBounds(220,460,95,30);

b2.setBounds(340,460,95,30);

reset.setBounds(460,460,95,30);

b1.setBackground(Color.WHITE);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

b4.setBackground(Color.WHITE);

b5.setBackground(Color.WHITE);

b5.setFont(font1);

reset.setBackground(Color.WHITE);

reset.setFont(font1);

b1.setFont(font1);

b2.setFont(font1);

b3.setFont(font1);

b4.setFont(font1);

f.add(reset);

f.add(l3);

f.add(b5);

f.add(l4);

f.add(l1);

f.add(tf1);

f.add(b3);

f.add(l2);

f.add(message);

f.add(message1);

f.add(message2);

f.add(b4);

f.add(b1);

f.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

reset.addActionListener(this);

f.setSize(750,550);

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\img1.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(30,30,750,550);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

f.add(lable);

f.setUndecorated(false);

f.setLocationRelativeTo(null);

f.setVisible(true);

f.setResizable(false);

}

private java.io.File showFileDialog(final boolean open) {

JFileChooser fc = new JFileChooser("Open an image");

javax.swing.filechooser.FileFilter ff = new javax.swing.filechooser.FileFilter() {

public boolean accept(java.io.File f) {

String name = f.getName().toLowerCase();

if(open)

return f.isDirectory() || name.endsWith(".jpg") || name.endsWith(".jpeg") ||

name.endsWith(".dib");

return f.isDirectory() || name.endsWith(".png");

}

public String getDescription() {

if(open)

return "Image (\*.jpg, \*.jpeg)";

return "Image (\*.png)";

}

};

fc.setAcceptAllFileFilterUsed(false);

fc.addChoosableFileFilter(ff);

java.io.File f = null;

if(open && fc.showOpenDialog(this) == fc.APPROVE\_OPTION)

f = fc.getSelectedFile();

else if(!open && fc.showSaveDialog(this) == fc.APPROVE\_OPTION)

f = fc.getSelectedFile();

File file = fc.getSelectedFile();

String fullPath = file.getAbsolutePath();

tf1.setText(fullPath);

return f;

}

private void openImage() {

java.io.File f = showFileDialog(true);

try {

sourceImage = ImageIO.read(f);

JLabel l = new JLabel(new ImageIcon(sourceImage));

originalPane.getViewport().add(l);

this.validate();

} catch(Exception ex) { ex.printStackTrace(); }

}

private void embedMessage()

{

String mess = message.getText();

embeddedImage = sourceImage.getSubimage(0,0,

sourceImage.getWidth(),sourceImage.getHeight());

embedMessage(embeddedImage, mess);

JLabel l = new JLabel(new ImageIcon(embeddedImage));

embeddedPane.getViewport().add(l);

this.validate();

addwatermark();

}

private void embedMessage(BufferedImage img, String mess) {

int messageLength = mess.length();

int imageWidth = img.getWidth(), imageHeight = img.getHeight(),

imageSize = imageWidth \* imageHeight;

if(messageLength \* 8 + 32 > imageSize) {

JOptionPane.showMessageDialog(this, "Message is too long for the chosen image",

"Message too long!", JOptionPane.ERROR\_MESSAGE);

return;

}

embedInteger(img, messageLength, 0, 0);

byte b[] = mess.getBytes();

for(int i=0; i<b.length; i++)

embedByte(img, b[i], i\*8+32, 0);

}

private void embedInteger(BufferedImage img, int n, int start, int storageBit) {

int maxX = img.getWidth(), maxY = img.getHeight(),

startX = start/maxY, startY = start - startX\*maxY, count=0;

for(int i=startX; i<maxX && count<32; i++) {

for(int j=startY; j<maxY && count<32; j++) {

int rgb = img.getRGB(i, j), bit = getBitValue(n, count);

rgb = setBitValue(rgb, storageBit, bit);

img.setRGB(i, j, rgb);

count++;

}

}

}

private void embedByte(BufferedImage img, byte b, int start, int storageBit) {

int maxX = img.getWidth(), maxY = img.getHeight(),

startX = start/maxY, startY = start - startX\*maxY, count=0;

for(int i=startX; i<maxX && count<8; i++) {

for(int j=startY; j<maxY && count<8; j++) {

int rgb = img.getRGB(i, j), bit = getBitValue(b, count);

rgb = setBitValue(rgb, storageBit, bit);

img.setRGB(i, j, rgb);

count++;

}

}

}

public void saveImage()

{

embedMessage();

if(embeddedImage == null)

{

JOptionPane.showMessageDialog(this, "No message has been embedded!",

"Nothing to save", JOptionPane.ERROR\_MESSAGE);

}

java.io.File f = showFileDialog(false);

String name = f.getName();

String ext = name.substring(name.lastIndexOf(".")+1).toLowerCase();

if(!ext.equals("png") && !ext.equals("dib")) {

ext = "png";

f = new java.io.File(f.getAbsolutePath()+".png");

String fullPath = f.getAbsolutePath();

message2.setText(fullPath);

addwatermark();

JOptionPane.showMessageDialog(null,"Secret Message has been embedded in image");

}

try {

if(f.exists()) f.delete();

ImageIO.write(embeddedImage, ext.toUpperCase(), f);

}

catch(Exception ex)

{

ex.printStackTrace();

}

}

public void check()

{

if(embeddedImage == null)

{

JOptionPane.showMessageDialog(null,"No Entries found");

}

else

{

f.dispose();

}

}

public void addwatermark()

{

BufferedImage bufferedImage = new BufferedImage(embeddedImage.getWidth(),

embeddedImage.getHeight(), BufferedImage.TYPE\_INT\_RGB);

Graphics graphics = bufferedImage.getGraphics();

graphics.drawImage(embeddedImage, 0, 0, null);

graphics.setFont(new Font("Times New Roman", Font.BOLD,40));

String watermark = message1.getText();

graphics.drawString(watermark,embeddedImage.getWidth()/3, embeddedImage.getHeight() / 3);

graphics.dispose();

embeddedImage= bufferedImage;

}

public void resetInterface()

{

f.dispose();

EmbedMessage em= new EmbedMessage();

}

private int getBitValue(int n, int location) {

int v = n & (int) Math.round(Math.pow(2, location));

return v==0?0:1;

}

private int setBitValue(int n, int location, int bit) {

int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);

if(bv == bit)

return n;

if(bv == 0 && bit == 1)

n |= toggle;

else if(bv == 1 && bit == 0)

n ^= toggle;

return n;

}

public void actionPerformed(ActionEvent ae) {

Object o = ae.getSource();

if(o==b1)

{

check();

}

else if(o == b2)

{

f.dispose();

}

if(o == b3)

{

openImage();

}

else if(o == b2)

{

embedMessage();

}

else if(o == b5)

{

saveImage();

}

else if(o == reset)

{

resetInterface();

}

}

}

class embed extends JFrame implements ActionListener

{

String path;

JLabel l1=new JLabel("AUDIO FILE(.wav) :");

JTextArea tf1=new JTextArea(10,3);;

JLabel l2=new JLabel(" TEXT FILE(.txt) :");

JTextArea tf2=new JTextArea(10,3);;

JLabel l3=new JLabel("OUTPUT FILE (.wav):");

JTextArea tf3=new JTextArea(10,3);;

JButton b1=new JButton("OK");

JButton b2=new JButton("CANCEL");

JButton b3=new JButton("BROWSE");

JButton b4=new JButton("BROWSE");

JButton b5=new JButton("BROWSE");

JFrame p=new JFrame();

public embed()

{

p.setSize(750,489);

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\blank1au.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(900,800,750,460);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

Font font = new Font("Calisto MT", Font.BOLD,20);

Font font1 = new Font("Calisto MT", Font.BOLD,13);

l1.setFont(font);

l2.setFont(font);

l3.setFont(font);

l1.setForeground(Color.WHITE);

l2.setForeground(Color.WHITE);

l3.setForeground(Color.WHITE);

l1.setBounds(112,110,198,40);

tf1.setBounds(313,119,115,25);

b3.setBounds(470,114,95,30);

l2.setBounds(110,160,218,40);

tf2.setBounds(313,166,115,25);

b4.setBounds(470,164,95,30);

l3.setBounds(91,210,218,40);

tf3.setBounds(313,217,115,25);

b5.setBounds(470,214,95,30);

b1.setBounds(260,270,95,30);

b2.setBounds(370,270,95,30);

b1.setBackground(Color.WHITE);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

b4.setBackground(Color.WHITE);

b5.setBackground(Color.WHITE);

b1.setFont(font1);

b2.setFont(font1);

b3.setFont(font1);

b4.setFont(font1);

b5.setFont(font1);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

b5.addActionListener(this);

p.add(l1);

p.add(tf1);

p.add(b3);

p.add(l2);

p.add(tf2);

p.add(b4);

p.add(l3);

p.add(tf3);

p.add(b5);

p.add(b1);

p.add(b2);

p.add(lable);

p.setUndecorated(false);

p.setLocationRelativeTo(null);

p.setVisible(true);

p.setResizable(false);

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if(e.getSource()==b3)

{

FileDialog fd=new FileDialog(this,"AUDIOFILE",FileDialog.LOAD);

fd.setVisible(true);

String f1=fd.getDirectory();

String f2=fd.getFile();

if((f2.substring(f2.lastIndexOf(".") + 1)).equals("wav"))

{

String fl=f1+f2;

tf1.setText(fl);

}

else

JOptionPane.showMessageDialog(null,"Please select the (.wav) audiofile");

}

else if(e.getSource()==b4)

{

FileDialog fd=new FileDialog(this,"TEXTFILE",FileDialog.LOAD);

fd.setVisible(true);

String f1=fd.getDirectory();

String f2=fd.getFile();

if((f2.substring(f2.lastIndexOf(".") + 1)).equals("txt") )

{

String fl=f1+f2;

tf2.setText(fl);

}

else

JOptionPane.showMessageDialog(null,"Please select the (.txt) textfile");

}

else if(e.getSource()==b5)

{

FileDialog fd=new FileDialog(this,"AUDIOFILE",FileDialog.SAVE);

fd.setVisible(true);

String f1=fd.getDirectory();

String f2=fd.getFile();

String fl=f1+f2;

tf3.setText(fl);

}

else if(e.getSource()==(b1))

{

int sizeaudio,sizetext,k,c,n;

String s1,s2,s3,out;

s1=tf1.getText();

s2=tf2.getText();

out=tf3.getText();

s3=out+".wav";

if(tf3.getText().equals("") || tf2.getText().equals("") || tf1.getText().equals(""))

JOptionPane.showMessageDialog(null,"Please enter the filenames");

else

{

try

{

File f=new File(s1);

//INPUT FILES AND OUTPUT FILES

FileInputStream faudio=new FileInputStream(f);

FileInputStream ftext=new FileInputStream(s2);

FileOutputStream emd=new FileOutputStream(s3);

//READ DATA FROM AUDIO FILE

sizeaudio = faudio.available();

int audio[]=new int[sizeaudio];

int audiodup[]=new int[sizeaudio];

int audiobits[][]=new int[sizeaudio][8];

for(int i=0;i<sizeaudio;i++)

{

audio[i]=faudio.read();

audiodup[i]=audio[i];

}

//READ DATA FROM TEXT FILE

sizetext= ftext.available();

int text[]=new int[sizetext];

c=0;

for(int i=0;i<sizetext;i++)

{

text[i]=ftext.read();

if(text[i]=='x')

c=c+1;

}

c=c\*2;

int sizetextdup=sizetext+c+2;

int textdup[]=new int[sizetextdup];

int textbits[][]=new int[sizetextdup][8];

n=0;

textdup[n]='x';

for(int i=0;i<sizetext;i++)

{

if(text[i]=='x')

textdup[++n]='x';

textdup[++n]=text[i];

}

textdup[++n]='x';

//COMPARE AUDIO FILE AND TEXT FILE

int j=150;

int count=0;

while(j<sizeaudio)

{

if(audio[j]>=0)

{

count=count+1;

j=j+5;

}

else

while(audio[j]<0)

j=j+1;

}

int t;

t=8\*sizetext;

if(count<t)

JOptionPane.showMessageDialog(null,"Audiofile not sufficient");

else

{

//convert audiobytes into 8bits

for(int i=150;i<sizeaudio;i++)

{

k=0;

if(audio[i]>=0)

{

while(k!=8)

{

if(audio[i]%2==0)

audiobits[i][k]=0;

else

{

audiobits[i][k]=1;

audio[i]=audio[i]-1;

}

audio[i]=audio[i]/2;

k=k+1;

}

}}

//convert textfilebytes into 8bits

for(int i=0;i<=n;i++)

{

k=0;

while(k<8)

{

if(textdup[i]%2==0)

textbits[i][k]=0;

else

{

textbits[i][k]=1;

textdup[i]=textdup[i]-1;

}

textdup[i]=textdup[i]/2;

k=k+1;

}

}

//embedding

j=150;

count=0;

int no=0;

while(j<sizeaudio)

{

if(audiodup[j]>=0)

{

audiobits[j][0]=textbits[no][count];

count=count+1;

j=j+5;

if(count>=8)

{

count=0;

no=no+1;

}

if(no > n)

break;

}

else

while(audiodup[j]<0)

j=j+1;

}

//converting data

for(int i=150;i<sizeaudio;i++)

if(audiodup[i]>=0)

{

audio[i]=0;

k=0;

while(k<8)

{

audio[i]=audio[i]+(audiobits[i][k]\*((int)(java.lang.Math.pow(2,k))));

k=k+1;

}

}

//store the data in to given filename

for(int i=0;i<sizeaudio;i++)

emd.write(audio[i]);

JOptionPane.showMessageDialog(null,"Sucessfully completed");

System.exit(0);

}}

catch(Exception exc)

{

}

}

}

else if(e.getSource()==(b2))

{

p.dispose();

}

}

}

class DecodeMessage1 extends JFrame implements ActionListener

{

String path;

JLabel l1=new JLabel(" IMAGE FILE :");

JButton open = new JButton("BROWSE");

JButton decode = new JButton("DECODE");

JButton reset = new JButton("RESET");

JButton cancel =new JButton("CANCEL");

JTextArea message = new JTextArea(10,3);

BufferedImage image = null;

JScrollPane imagePane = new JScrollPane();

JFrame f =new JFrame(" ");

JTextArea tf1=new JTextArea(10,10);

public DecodeMessage1()

{

f.setSize(750,550);

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\img2.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(30,30,750,550);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

Font font = new Font("Calisto MT", Font.BOLD,20);

Font font1 = new Font("Calisto MT", Font.BOLD,13);

l1.setFont(font);

l1.setForeground(Color.WHITE);

l1.setBounds(112,110,198,40);

tf1.setBounds(313,119,140,25);

decode.setBounds(272,184,198,40);

open.setBounds(470,114,95,30);

message.setBounds(172,250,400,100);

cancel.setBounds(400,380,95,30);

reset.setBounds(260,380,95,30);

f.add(reset);

f.add(message);

f.add(l1);

f.add(open);

f.add(decode);

f.add(tf1);

f.add(cancel);

open.setFont(font1);

decode.setFont(font1);

reset.setFont(font1);

cancel.setFont(font1);

cancel.setBackground(Color.WHITE);

reset.setBackground(Color.WHITE);

open.setBackground(Color.WHITE);

decode.setBackground(Color.WHITE);

f.add(lable);

f.setUndecorated(false);

f.setLocationRelativeTo(null);

f.setVisible(true);

f.setResizable(false);

open.addActionListener(this);

decode.addActionListener(this);

reset.addActionListener(this);

cancel.addActionListener(this);

}

public void actionPerformed(ActionEvent ae)

{

Object o = ae.getSource();

if(o==cancel)

{

f.dispose();

}

else if(o==reset)

{

f.dispose();

DecodeMessage1 dm=new DecodeMessage1();

}

else if(o == open)

{

openImage();

}

else if(o == decode)

{

decodeMessage1();

}

else if(o == reset)

{

resetInterface();

}

}

private java.io.File showFileDialog(boolean open)

{

JFileChooser fc = new JFileChooser("Open an image");

javax.swing.filechooser.FileFilter ff = new javax.swing.filechooser.FileFilter() {

public boolean accept(java.io.File f) {

String name = f.getName().toLowerCase();

return f.isDirectory() || name.endsWith(".png");

}

public String getDescription() {

return "Image (\*.png)";

}

};

fc.setAcceptAllFileFilterUsed(false);

fc.addChoosableFileFilter(ff);

java.io.File f = null;

if(open && fc.showOpenDialog(this) == fc.APPROVE\_OPTION)

f = fc.getSelectedFile();

else if(!open && fc.showSaveDialog(this) == fc.APPROVE\_OPTION)

f = fc.getSelectedFile();

return f;

}

private void openImage() {

java.io.File f = showFileDialog(true);

try {

image = ImageIO.read(f);

JLabel l = new JLabel(new ImageIcon(image));

imagePane.getViewport().add(l);

String fullPath = f.getAbsolutePath();

tf1.setText(fullPath);

this.validate();

}

catch(Exception ex)

{

ex.printStackTrace();

}

}

private void decodeMessage1()

{

int len = extractInteger(image, 0, 0);

byte b[] = new byte[len];

for(int i=0; i<len; i++)

b[i] = extractByte(image, i\*8+32, 0);

message.setText(new String(b));

}

private int extractInteger(BufferedImage img, int start, int storageBit)

{

int maxX = img.getWidth(), maxY = img.getHeight(),

startX = start/maxY, startY = start - startX\*maxY, count=0;

int length = 0;

for(int i=startX; i<maxX && count<32; i++) {

for(int j=startY; j<maxY && count<32; j++) {

int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);

length = setBitValue(length, count, bit);

count++;

}

}

return length;

}

private byte extractByte(BufferedImage img, int start, int storageBit)

{

int maxX = img.getWidth(), maxY = img.getHeight(),

startX = start/maxY, startY = start - startX\*maxY, count=0;

byte b = 0;

for(int i=startX; i<maxX && count<8; i++) {

for(int j=startY; j<maxY && count<8; j++) {

int rgb = img.getRGB(i, j), bit = getBitValue(rgb, storageBit);

b = (byte)setBitValue(b, count, bit);

count++;

}

}

return b;

}

private void resetInterface()

{

message.setText("");

imagePane.getViewport().removeAll();

image = null;

this.validate();

}

private int getBitValue(int n, int location)

{

int v = n & (int) Math.round(Math.pow(2, location));

return v==0?0:1;

}

private int setBitValue(int n, int location, int bit)

{

int toggle = (int) Math.pow(2, location), bv = getBitValue(n, location);

if(bv == bit)

return n;

if(bv == 0 && bit == 1)

n |= toggle;

else if(bv == 1 && bit == 0)

n ^= toggle;

return n;

}

}

class home1 extends JFrame implements ActionListener

{

JFrame f=new JFrame(" AUDIO STEGANOGRAPHY");

String path1="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\audio.jpg";

public home1()

{

JButton b1=new JButton("EMBED");

JButton b2=new JButton("EXTRACT");

JButton b3=new JButton(" BACK ");

f.setSize(750,489);

ImageIcon ii = new ImageIcon(path1);

JLabel lable = new JLabel();

lable.setBounds(0,0,1371,755);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

f.add(lable);

b1.setBounds(242,550,290,50);

b2.setBounds(821,550,290,50);

b3.setBounds(530,650,290,50);

b1.setBackground(Color.WHITE);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

Font font = new Font("Calisto MT", Font.BOLD,20);

b1.setFont(font);

b2.setFont(font);

b3.setFont(font);

f.add(b1);

f.add(b2);

f.add(b3);

f.setExtendedState(Frame.MAXIMIZED\_BOTH);

f.setUndecorated(false);

f.setLayout(null);

f.add(lable);

f.setVisible(true);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if(s.equals("EMBED"))

{

embed em=new embed();

}

else if(s.equals("EXTRACT"))

{

extract x=new extract();

}

else if(s.equals(" BACK "))

{

f.dispose();

}

}

class extract extends JFrame implements ActionListener

{

String path;

JLabel l1=new JLabel(" AUDIO FILE (.wav) :");

JTextArea tf1=new JTextArea(10,3);;

JLabel l2=new JLabel(" TEXT FILE (.txt) :");

JTextArea tf2=new JTextArea(10,3);;

JButton b1=new JButton("OK");

JButton b2=new JButton("CANCEL");

JButton b3=new JButton("BROWSE");

JButton b4=new JButton("BROWSE");

JFrame f=new JFrame(" EXTRACT TEXT FROM AUDIO");

public extract()

{

Font font = new Font("Calisto MT", Font.BOLD,20);

Font font1 = new Font("Calisto MT", Font.BOLD,13);

l1.setFont(font);

l2.setFont(font);

l1.setForeground(Color.WHITE);

l2.setForeground(Color.WHITE);

l1.setBounds(112,110,198,40);

tf1.setBounds(313,119,115,25);

b3.setBounds(470,114,95,30);

l2.setBounds(110,160,218,40);

tf2.setBounds(313,166,115,25);

b4.setBounds(470,164,95,30);

b1.setBounds(250,240,95,30);

b2.setBounds(370,240,95,30);

b1.setBackground(Color.WHITE);

b2.setBackground(Color.WHITE);

b3.setBackground(Color.WHITE);

b4.setBackground(Color.WHITE);

b1.setFont(font1);

b2.setFont(font1);

b3.setFont(font1);

b4.setFont(font1);

f.add(l1);

f.add(tf1);

f.add(b3);

f.add(l2);

f.add(tf2);

f.add(b4);

f.add(b1);

f.add(b2);

b1.addActionListener(this);

b2.addActionListener(this);

b3.addActionListener(this);

b4.addActionListener(this);

f.setSize(750,489);

path="C:\\Users\\OM KHURANA\\Documents\\NetBeansProjects\\steganography\\images\\blank2au.jpg";

ImageIcon ii = new ImageIcon(path);

JLabel lable = new JLabel();

lable.setBounds(30,30,750,460);

Image img= ii.getImage();

Image newImg=img.getScaledInstance(lable.getWidth(),lable.getHeight(),Image.SCALE\_SMOOTH);

ImageIcon image=new ImageIcon(newImg);

lable.setIcon(image);

f.add(lable);

f.setUndecorated(false);

f.setLocationRelativeTo(null);

f.setVisible(true);

f.setResizable(false);

}

public void actionPerformed(ActionEvent e)

{

String s=e.getActionCommand();

if(e.getSource()==b3)

{

FileDialog fd=new FileDialog(this,"AUDIOFILE",FileDialog.LOAD);

fd.setVisible(true);

String f1=fd.getDirectory();

String f2=fd.getFile();

if((f2.substring(f2.lastIndexOf(".") + 1)).equals("wav"))

{

String fl=f1+f2;

tf1.setText(fl);

}

else

JOptionPane.showMessageDialog(null,"Please select the (.wav) audiofile");

}

else if(e.getSource()==b4)

{

FileDialog fd=new FileDialog(this,"TEXTFILE",FileDialog.SAVE);

fd.setVisible(true);

String f1=fd.getDirectory();

String f2=fd.getFile();

String fl=f1+f2;

tf2.setText(fl);

}

else if(e.getSource()==(b1))

{

int c,n,k;

String s1,s2,out;

s1=tf1.getText();

out=tf2.getText();

s2=out+".txt";

if(tf2.getText().equals("") || tf1.getText().equals(""))

JOptionPane.showMessageDialog(null,"Please enter the filenames");

else

{

try

{

//INPUT FILES AND OUTPUT FILES

InputStream faudio=new FileInputStream(s1);

OutputStream txt=new FileOutputStream(s2);

//READ DATA FROM AUDIO FILE

int sizeaudio = faudio.available();

int audio[]=new int[sizeaudio];

int audiobits[][]=new int[sizeaudio][8];

for(int i=0;i<sizeaudio;i++)

{

audio[i]=faudio.read();

}

c=0;

//convert audiobytes into 8bits

for(int i=150;i<sizeaudio;i++)

{

k=0;

if(audio[i]>=0)

{

while(k!=8)

{

if(audio[i]%2==0)

audiobits[i][k]=0;

else

{

audiobits[i][k]=1;

audio[i]=audio[i]-1;

}

audio[i]=audio[i]/2;

k=k+1;

}

c=c+1;

}}

n=(int)(java.lang.Math.ceil(c/8));

int textbits[][]=new int[c][8];

int text[]=new int[n];

//embedding

int j=150;

int count=0;

int no=0;

while(j<sizeaudio)

{

if(audio[j]>=0)

{

textbits[no][count]=audiobits[j][0];

count=count+1;

j=j+5;

if(count>=8)

{

count=0;

no=no+1;

}

if(no > n)

break;

}

else

while(audio[j]<0)

j=j+1;

}

//converting data

for(int i=0;i<n;i++)

{

text[i]=0;

k=0;

while(k<8)

{

text[i]=text[i]+(textbits[i][k]\*((int)(java.lang.Math.pow(2,k))));

k=k+1;

}

}

for(int i=1;i<n;i++)

{

if(text[0]=='x')

{

if(text[i]=='x'&&text[i+1]=='x')

{

txt.write(text[i]);

i=i+1;

}

else if(text[i]=='x')

break;

else

txt.write(text[i]);

}}

File ff=new File(s2);

InputStream fis=new FileInputStream(ff);

if(fis.available()==0)

{

ff.delete();

JOptionPane.showMessageDialog(null,"Please enter the correct audiofile");

}

else

{

JOptionPane.showMessageDialog(null,"Sucessfully completed");

}

}

catch(Exception exc)

{

}

}}

else if(e.getSource()==(b2))

{

f.dispose();

}

}

}

}