Міністерство освіти і науки України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ



Звіт

з лабораторної роботи № 4

з дисципліни «Захист інформації в комп’ютерних системах»

на тему: «Організація стеганоканалу в BMP-файлі»

Виконав: ст. гр. КІ-302

Радевич-Винницький Я.А.

Перевірив:

Муляревич О.В.

**Мета роботи:** ознайомитися з поняттям стеганографії та проаналізувати можливості організації стеганоканалу в BMP-файлі.

**Завдання:**

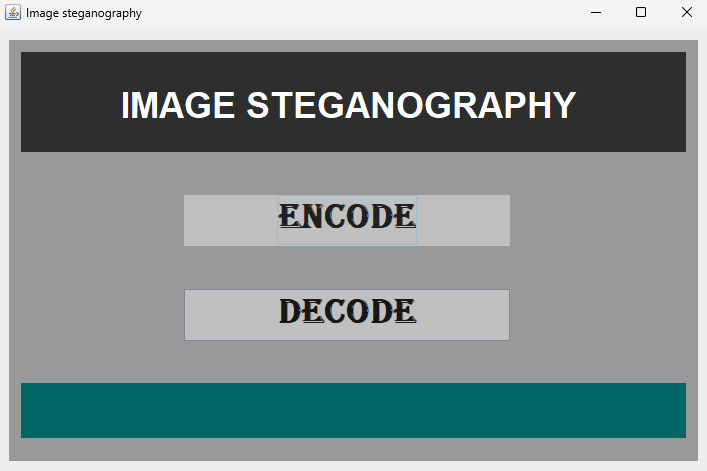
Проаналізувавши формат ВМР, створити програму для організації стегоканалу для приховування даних, що в середньому є в 10 раз меншими за розмір файлу.

**Варіант:** 24

**Виконання завдання:**

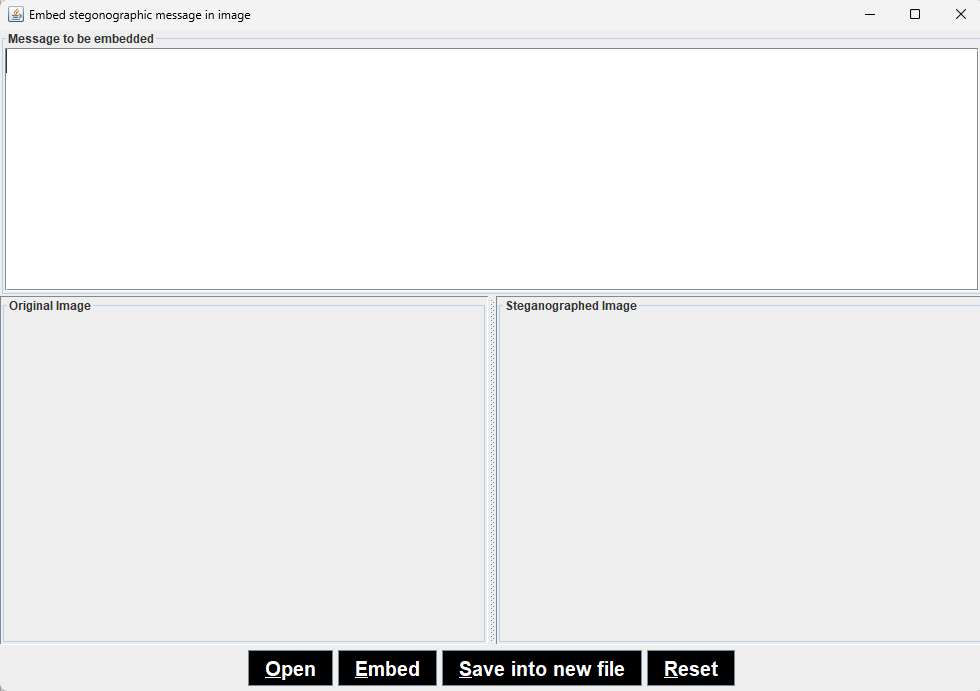
Для виконання завдання було вибрано мову Java та бібліотеку Swing для створення графічного інтерфейсу. Програмний код наведено в додатку.

Демонстрація роботи програми:

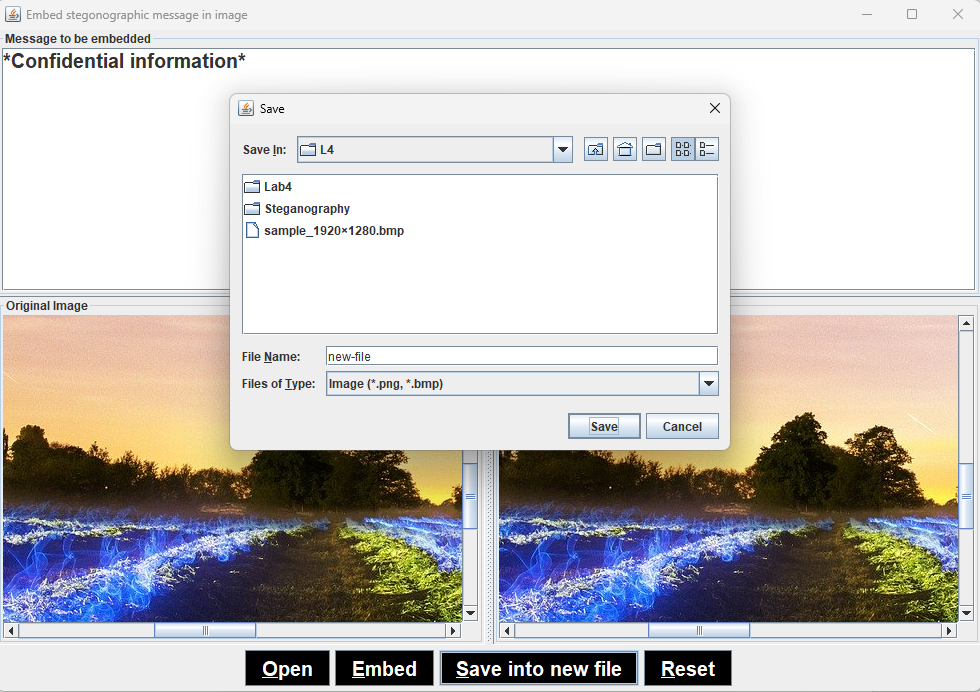


*Рис. 1 – початкове вікно програми*

Перше вікно програми пропонує користувачеві на вибір дві функції: кодування та розкодування повідомлень. Після вибору функції “Encode” відкривається наступне вікно. У ньому, за допомогою кнопки Open, можна вибрати файл-контейнер, а також ввести інформацію у текстове поле, провести кодування за допомогою кнопки Embed, зберегти закодований файл за допомогою кнопки Save into new file.

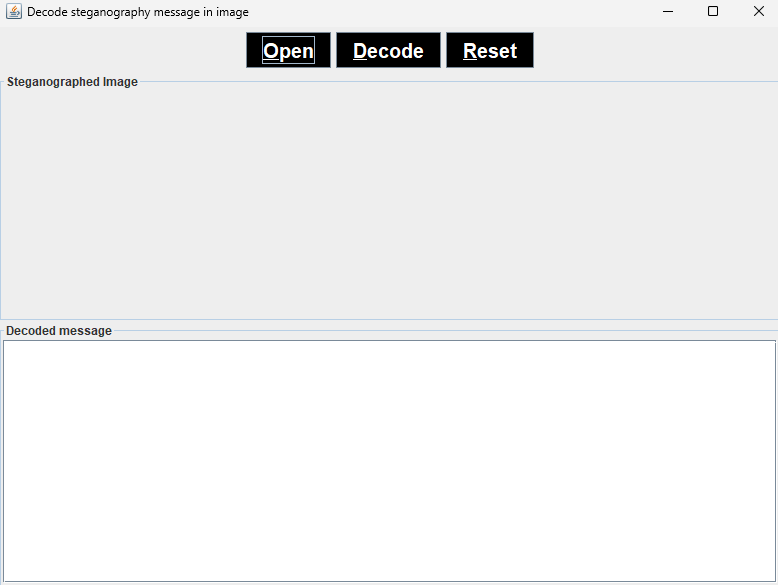


*Рис. 2 – основне вікно програми в режимі “Encode”*

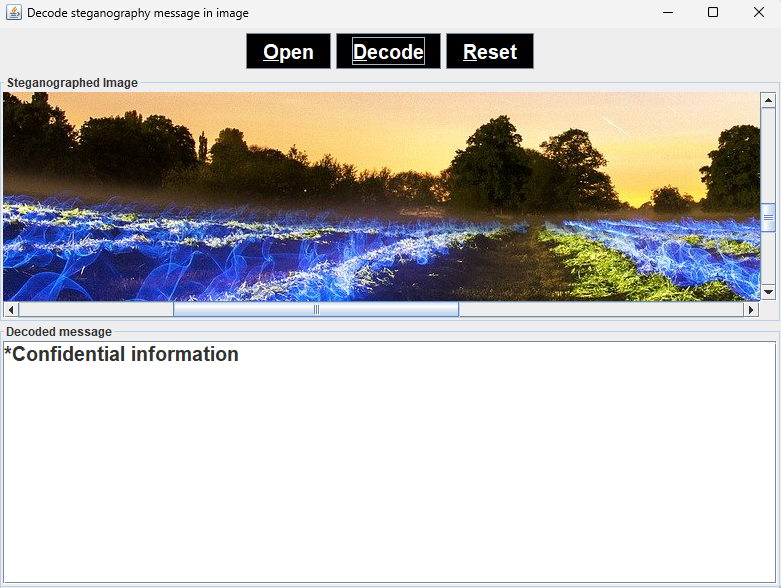


*Рис. 3 – основне вікно програми з даними*

Вибравши функцію “Decode” користувач побачить наступне вікно. У ньому за допомогою кнопки Open можна вибрати файл, який містить приховане повідомлення. Отримати його можна використавши кнопку Decode.



*Рис. 4 – основне вікно програми в режимі “Decode”*



*Рис. 5 – основне вікно програми в режимі “Decode” з даними*

**Висновок:** у ході виконання лабораторної роботи було вивчено поняття стеганографії, проаналізовано можливості організації стеганоканалу в BMP-файлі та створено прикладну програму для приховування повідомлення.

**Додаток**

Код файлу *Encryption.java*:

Лістинг 1

|  |
| --- |
| import javax.imageio.ImageIO; import javax.swing.BorderFactory; import javax.swing.ImageIcon; import javax.swing.JButton; import javax.swing.JFileChooser; import javax.swing.JFrame; import javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JPanel; import javax.swing.JScrollPane; import javax.swing.JSplitPane; import javax.swing.JTextArea; import javax.swing.filechooser.FileFilter; import java.awt.BorderLayout; import java.awt.Color; import java.awt.FlowLayout; import java.awt.Font; import java.awt.GridLayout; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.image.BufferedImage;  public class Encryption extends JFrame implements ActionListener {  JButton open = new JButton("Open");  JButton embed = new JButton("Embed");  JButton save = new JButton("Save into new file");  JButton reset = new JButton("Reset");   JTextArea message = new JTextArea(10, 3);  BufferedImage sourceImage = null, embeddedImage = null;  JSplitPane sp = new JSplitPane(JSplitPane.*HORIZONTAL\_SPLIT*);  JScrollPane originalPane = new JScrollPane(),  embeddedPane = new JScrollPane();   public Encryption() {  super("Embed stegonographic message in image");  assembleInterface();  open.setBackground(Color.*black*);  open.setForeground(Color.*WHITE*);  open.setFont(new Font("Monaco", Font.*BOLD*, 20));   embed.setBackground(Color.*black*);  embed.setForeground(Color.*WHITE*);  embed.setFont(new Font("Monaco", Font.*BOLD*, 20));   save.setBackground(Color.*black*);  save.setForeground(Color.*WHITE*);  save.setFont(new Font("Monaco", Font.*BOLD*, 20));   reset.setBackground(Color.*black*);  reset.setForeground(Color.*WHITE*);  reset.setFont(new Font("Monaco", Font.*BOLD*, 20));   this.setSize(1000, 700);  this.setLocationRelativeTo(null);  this.setDefaultCloseOperation(*DISPOSE\_ON\_CLOSE*);  this.setVisible(true);  sp.setDividerLocation(0.5);  this.validate();  }   private void assembleInterface() {  JPanel p = new JPanel(new FlowLayout());  p.add(open);  p.add(embed);  p.add(save);  p.add(reset);  this.getContentPane().add(p, BorderLayout.*SOUTH*);  open.addActionListener(this);  embed.addActionListener(this);  save.addActionListener(this);  reset.addActionListener(this);  open.setMnemonic('O');  embed.setMnemonic('E');  save.setMnemonic('S');  reset.setMnemonic('R');   p = new JPanel(new GridLayout(1, 1));  p.add(new JScrollPane(message));  message.setFont(new Font("Arial", Font.*BOLD*, 20));  p.setBorder(BorderFactory.*createTitledBorder*("Message to be embedded"));  this.getContentPane().add(p, BorderLayout.*NORTH*);   sp.setLeftComponent(originalPane);  sp.setRightComponent(embeddedPane);  originalPane.setBorder(BorderFactory.*createTitledBorder*("Original Image"));  embeddedPane.setBorder(BorderFactory.*createTitledBorder*("Steganographed Image"));  this.getContentPane().add(sp, BorderLayout.*CENTER*);  }   public void actionPerformed(ActionEvent ae) {  Object o = ae.getSource();  if (o == open)  openImage();  else if (o == embed)  embedMessage();  else if (o == save)  saveImage();  else if (o == reset)  resetInterface();  }   private java.io.File showFileDialog(final boolean open) {  JFileChooser fc = new JFileChooser("Open an image");  FileFilter ff = new 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(".png") || name.endsWith(".gif") || name.endsWith(".tiff") ||  name.endsWith(".bmp") || name.endsWith(".dib");  return f.isDirectory() || name.endsWith(".png") || name.endsWith(".bmp");  }   public String getDescription() {  if (open)  return "Image (\*.jpg, \*.jpeg, \*.png, \*.gif, \*.tiff, \*.bmp, \*.dib)";  return "Image (\*.png, \*.bmp)";  }  };  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 {  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();  }   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++;  }  }  }   private void saveImage() {  if (embeddedImage == null) {  JOptionPane.*showMessageDialog*(this, "No message has been embedded!",  "Nothing to save", JOptionPane.*ERROR\_MESSAGE*);  return;  }  java.io.File f = showFileDialog(false);  String name = f.getName();  String ext = name.substring(name.lastIndexOf(".") + 1).toLowerCase();  if (!ext.equals("png") && !ext.equals("bmp") && !ext.equals("dib")) {  ext = "png";  f = new java.io.File(f.getAbsolutePath() + ".png");  }  try {  if (f.exists()) f.delete();  ImageIO.*write*(embeddedImage, ext.toUpperCase(), f);  } catch (Exception ex) {  ex.printStackTrace();  }  }   private void resetInterface() {  message.setText("");  originalPane.getViewport().removeAll();  embeddedPane.getViewport().removeAll();  sourceImage = null;  embeddedImage = null;  sp.setDividerLocation(0.5);  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;  } } |

Код файлу *Decryption.java*:

Лістинг 2

|  |
| --- |
| import javax.imageio.ImageIO; import javax.swing.BorderFactory; import javax.swing.ImageIcon; import javax.swing.JButton; import javax.swing.JFileChooser; import javax.swing.JFrame; import javax.swing.JLabel; import javax.swing.JOptionPane; import javax.swing.JPanel; import javax.swing.JScrollPane; import javax.swing.JTextArea; import javax.swing.filechooser.FileFilter; import java.awt.BorderLayout; import java.awt.Color; import java.awt.FlowLayout; import java.awt.Font; import java.awt.GridLayout; import java.awt.event.ActionEvent; import java.awt.event.ActionListener; import java.awt.image.BufferedImage; import java.io.File;  public class Decryption extends JFrame implements ActionListener {  JButton open = new JButton("Open");  JButton decode = new JButton("Decode");  JButton reset = new JButton("Reset");  JTextArea message = new JTextArea(10, 3);  BufferedImage image = null;  JScrollPane imagePane = new JScrollPane();   public Decryption() {  super("Decode steganography message in image");  assembleInterface();  this.setSize(800, 600);  this.setLocationRelativeTo(null);  this.setDefaultCloseOperation(*DISPOSE\_ON\_CLOSE*);  this.setVisible(true);   open.setBackground(Color.*black*);  open.setForeground(Color.*WHITE*);  open.setFont(new Font("Monaco", Font.*BOLD*, 20));   decode.setBackground(Color.*black*);  decode.setForeground(Color.*WHITE*);  decode.setFont(new Font("Monaco", Font.*BOLD*, 20));   reset.setBackground(Color.*black*);  reset.setForeground(Color.*WHITE*);  reset.setFont(new Font("Monaco", Font.*BOLD*, 20));   }   private void assembleInterface() {  JPanel p = new JPanel(new FlowLayout());  p.add(open);  p.add(decode);  p.add(reset);  this.getContentPane().add(p, BorderLayout.*NORTH*);  open.addActionListener(this);  decode.addActionListener(this);  reset.addActionListener(this);  open.setMnemonic('O');  decode.setMnemonic('D');  reset.setMnemonic('R');   p = new JPanel(new GridLayout(1, 1));  p.add(new JScrollPane(message));  message.setFont(new Font("Arial", Font.*BOLD*, 20));  p.setBorder(BorderFactory.*createTitledBorder*("Decoded message"));  message.setEditable(false);  this.getContentPane().add(p, BorderLayout.*SOUTH*);   imagePane.setBorder(BorderFactory.*createTitledBorder*("Steganographed Image"));  this.getContentPane().add(imagePane, BorderLayout.*CENTER*);  }   public void actionPerformed(ActionEvent ae) {  Object o = ae.getSource();  if (o == open)  openImage();  else if (o == decode)  decodeMessage();  else if (o == reset)  resetInterface();  }   private File showFileDialog(boolean open) {  JFileChooser fc = new JFileChooser("Open an image");  FileFilter ff = new FileFilter() {  public boolean accept(java.io.File f) {  String name = f.getName().toLowerCase();  return f.isDirectory() || name.endsWith(".png") || name.endsWith(".bmp");  }   public String getDescription() {  return "Image (\*.png, \*.bmp)";  }  };  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() {  File f = showFileDialog(true);  try {  image = ImageIO.*read*(f);  JLabel l = new JLabel(new ImageIcon(image));  imagePane.getViewport().add(l);  this.validate();  } catch (Exception ex) {  ex.printStackTrace();  }  }   private void decodeMessage() {  if (image == null) {  JOptionPane.*showMessageDialog*(null, "first open a picture");  return;  }  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;  }  } |

Код файлу *MenuFrame.java*:

Лістинг 3

|  |
| --- |
| import javax.swing.BorderFactory; import javax.swing.GroupLayout; import javax.swing.JButton; import javax.swing.JFrame; import javax.swing.JLabel; import javax.swing.JPanel; import javax.swing.LayoutStyle; import javax.swing.SwingConstants; import javax.swing.WindowConstants; import java.awt.Color; import java.awt.Cursor; import java.awt.Dimension; import java.awt.Font; import java.awt.event.ActionEvent; import java.awt.event.MouseAdapter; import java.awt.event.MouseEvent;  public class MenuFrame extends JFrame {   public MenuFrame() {  initComponents();  }    @SuppressWarnings("unchecked")  private void initComponents() {  jPanel2 = new JPanel();  jButton2 = new JButton();  jButton1 = new JButton();  jPanel1 = new JPanel();  jLabel1 = new JLabel();  jPanel3 = new JPanel();  jLabel2 = new JLabel();   setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  setTitle("Image steganography");   jPanel2.setBackground(new Color(153, 153, 153));   jButton2.setBackground(new Color(223, 223, 223));  jButton2.setFont(new Font("Algerian", 1, 36)); jButton2.setForeground(new Color(32, 32, 32));  jButton2.setText("ENCODE");  jButton2.setBorder(BorderFactory.*createEmptyBorder*(1, 1, 1, 1));  jButton2.setCursor(new Cursor(Cursor.*HAND\_CURSOR*));  jButton2.addMouseListener(new MouseAdapter() {  public void mouseEntered(MouseEvent evt) {  jButton2MouseEntered(evt);  }  });  jButton2.addActionListener(this::jButton2ActionPerformed);   jButton1.setBackground(new Color(224, 227, 225));  jButton1.setFont(new Font("Algerian", 1, 36)); jButton1.setForeground(new Color(23, 23, 23));  jButton1.setText("DECODE");  jButton1.setCursor(new Cursor(Cursor.*HAND\_CURSOR*));  jButton1.addMouseListener(new MouseAdapter() {  public void mouseEntered(MouseEvent evt) {  jButton1MouseEntered(evt);  }  });  jButton1.addActionListener(this::jButton1ActionPerformed);   jPanel1.setBackground(new Color(46, 46, 46));   jLabel1.setFont(new Font("Sitka Banner", 1, 36)); jLabel1.setForeground(new Color(254, 254, 254));  jLabel1.setHorizontalAlignment(SwingConstants.*CENTER*);  jLabel1.setText("IMAGE STEGANOGRAPHY ");   GroupLayout jPanel1Layout = new GroupLayout(jPanel1);  jPanel1.setLayout(jPanel1Layout);  jPanel1Layout.setHorizontalGroup(  jPanel1Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(GroupLayout.Alignment.*TRAILING*, jPanel1Layout.createSequentialGroup()  .addContainerGap()  .addComponent(jLabel1, GroupLayout.*DEFAULT\_SIZE*, 653, Short.*MAX\_VALUE*)  .addContainerGap())  );  jPanel1Layout.setVerticalGroup(  jPanel1Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(jPanel1Layout.createSequentialGroup()  .addGap(26, 26, 26)  .addComponent(jLabel1, GroupLayout.*PREFERRED\_SIZE*, 52, GroupLayout.*PREFERRED\_SIZE*)  .addContainerGap(22, Short.*MAX\_VALUE*))  );   jPanel3.setBackground(new Color(0, 102, 102));   jLabel2.setFont(new Font("Tahoma", 1, 18)); jLabel2.setForeground(new Color(255, 255, 255));  jLabel2.setHorizontalAlignment(SwingConstants.*CENTER*);   GroupLayout jPanel3Layout = new GroupLayout(jPanel3);  jPanel3.setLayout(jPanel3Layout);  jPanel3Layout.setHorizontalGroup(  jPanel3Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(jPanel3Layout.createSequentialGroup()  .addContainerGap()  .addComponent(jLabel2, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)  .addContainerGap())  );  jPanel3Layout.setVerticalGroup(  jPanel3Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(GroupLayout.Alignment.*TRAILING*, jPanel3Layout.createSequentialGroup()  .addContainerGap(GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)  .addComponent(jLabel2, GroupLayout.*PREFERRED\_SIZE*, 37, GroupLayout.*PREFERRED\_SIZE*)  .addGap(27, 27, 27))  );   GroupLayout jPanel2Layout = new GroupLayout(jPanel2);  jPanel2.setLayout(jPanel2Layout);  jPanel2Layout.setHorizontalGroup(  jPanel2Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(jPanel2Layout.createSequentialGroup()  .addGap(175, 175, 175)  .addGroup(jPanel2Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*, false)  .addComponent(jButton2, GroupLayout.*DEFAULT\_SIZE*, 326, Short.*MAX\_VALUE*)  .addComponent(jButton1, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))  .addGap(0, 0, Short.*MAX\_VALUE*))  .addGroup(jPanel2Layout.createSequentialGroup()  .addContainerGap()  .addGroup(jPanel2Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addComponent(jPanel1, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)  .addComponent(jPanel3, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*))  .addContainerGap())  );  jPanel2Layout.setVerticalGroup(  jPanel2Layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(jPanel2Layout.createSequentialGroup()  .addContainerGap()  .addComponent(jPanel1,GroupLayout.*PREFERRED\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*PREFERRED\_SIZE*)  .addGap(43, 43, 43)  .addComponent(jButton2, GroupLayout.*PREFERRED\_SIZE*, 51, GroupLayout.*PREFERRED\_SIZE*)  .addGap(43, 43, 43)  .addComponent(jButton1, GroupLayout.*PREFERRED\_SIZE*, 52, GroupLayout.*PREFERRED\_SIZE*)  .addPreferredGap(LayoutStyle.ComponentPlacement.*RELATED*, 45, Short.*MAX\_VALUE*)  .addComponent(jPanel3, GroupLayout.*PREFERRED\_SIZE*, 55, GroupLayout.*PREFERRED\_SIZE*)  .addGap(23, 23, 23))  );   GroupLayout layout = new GroupLayout(getContentPane());  getContentPane().setLayout(layout);  layout.setHorizontalGroup(  layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(GroupLayout.Alignment.*TRAILING*, layout.createSequentialGroup()  .addContainerGap()  .addComponent(jPanel2, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)  .addContainerGap())  );  layout.setVerticalGroup(  layout.createParallelGroup(GroupLayout.Alignment.*LEADING*)  .addGroup(GroupLayout.Alignment.*TRAILING*, layout.createSequentialGroup()  .addContainerGap()  .addComponent(jPanel2, GroupLayout.*DEFAULT\_SIZE*, GroupLayout.*DEFAULT\_SIZE*, Short.*MAX\_VALUE*)  .addContainerGap())  );   setSize(new Dimension(729, 484));  setVisible(true);  setLocationRelativeTo(null);  }   private void jButton2ActionPerformed(ActionEvent evt) {  Encryption a=new Encryption();  a.setVisible(true);  }   private void jButton1ActionPerformed(ActionEvent evt) {  Decryption a=new Decryption();    a.setVisible(true);  }   private void jButton2MouseEntered(MouseEvent evt) {   jButton2.setBackground(Color.*lightGray*);  }   private void jButton1MouseEntered(MouseEvent evt) {   jButton1.setBackground(Color.*LIGHT\_GRAY*);  }   private JButton jButton1;  private JButton jButton2;  private JLabel jLabel1;  private JLabel jLabel2;  private JPanel jPanel1;  private JPanel jPanel2;  private JPanel jPanel3;  } |

Код файлу *Main.java:*

Лістинг 4

|  |
| --- |
| public class Main {  public static void main(String[] args) {  MenuFrame menuFrame = new MenuFrame();  } } |