Surname & Initials: Basopu A

Student number: 230172234

Module: Advance Object Oriented Programming

Course code: AOP 216D

Task: Assignment 2

Topic: Graphic User Interface

Due date: 12 August 2024

Question 1: CRECHE APPLICATION

```
package za.ac.tut.gui;

import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author Alex
 */
public class CreshLibrary extends JFrame
{
   //DECLARING VARIABLES
   private JPanel namePnel;
   private JPanel genderPanel;
   private JPanel  buttonPanel;
   private JPanel  mainPnel;
   private JPanel areaPanel;
   private JLabel nameLabel;
   private JLabel genderLabel;
   private JTextField nameField;
   private JTextArea  area;
   private JRadioButton male;
```

```
private JRadioButton female;
private JButton register;
private JButton display;
private JScrollPane scrollbar;
private List<Child> kids = new ArrayList<>();

public CreshLibrary()
{
   //CREATING  FRAME
      setSize(400, 400);
      setLayout(new BorderLayout());
      setTitle("CRECHE 4 YOUR KIDDIE");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

   // Initializing
      namePnel = new JPanel(new FlowLayout(FlowLayout.LEFT));
      genderPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));
      buttonPanel = new JPanel(new FlowLayout(FlowLayout.LEFT));
      mainPnel = new JPanel(new BorderLayout());
      areaPanel = new JPanel(new BorderLayout());
   //TEXT FIELD
      nameField = new JTextField(15);

   // TEXT AREA
      area = new JTextArea(10, 30);
      nameField = new JTextField(15);
      area = new JTextArea(10, 30);
      area.setEditable(false);
      scrollbar = new JScrollPane(area);
      scrollbar.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
      scrollbar.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR_ALWAYS);
```

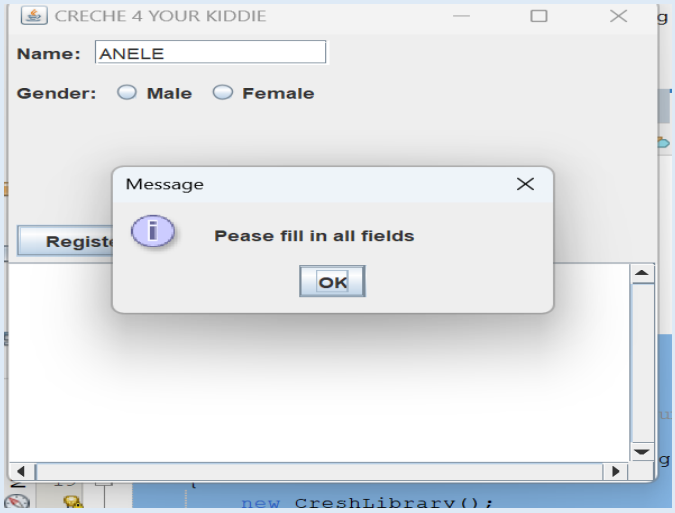
```
    //LABELS
        nameLabel = new JLabel("Name: ");
        genderLabel = new JLabel("Gender: ");

//CREATING RADIO BUTTONS
        male = new JRadioButton("Male");
        female = new JRadioButton("Female");
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(male);
        genderGroup.add(female);

// CREATING BUTTONS
        register = new JButton("Register kiddie");
        display = new JButton("Display kiddie");

// Adding components to panels
        namePnel.add(nameLabel);
        namePnel.add(nameField);
        genderPanel.add(genderLabel);
        genderPanel.add(male);
        genderPanel.add(female);
        buttonPanel.add(register);
        buttonPanel.add(display);
        areaPanel.add(scrollbar, BorderLayout.CENTER);

//ADDING TO THE MAIN PANEL
        mainPnel.add(namePnel, BorderLayout.NORTH);
        mainPnel.add(genderPanel, BorderLayout.CENTER);
        mainPnel.add(buttonPanel, BorderLayout.SOUTH);
```

```
        add(mainPnel, BorderLayout.CENTER);
        add(areaPanel, BorderLayout.SOUTH);

    // Adding action listeners
        register.addActionListener(new RegisterButton());
       display.addActionListener(new DisplayButton());

        setVisible(true);



    }
    //CAPTURING METHOD
    private class RegisterButton implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            String name = nameField.getText();
            String genders = male.isSelected()? "Male" : female.isSelected() ? "Female " : null;

            if(name.isEmpty() || genders == null)
            {
                JOptionPane.showMessageDialog(CreshLibrary.this," Pease fill in all fields");
                return;
            }
            Child child = new Child(name,genders);
            kids.add(child);
            nameField.setText("");
            male.setSelected(false);
            female.setSelected(false);
```

```
            JOptionPane.showMessageDialog(CreshLibrary.this," kid is Registered Successfully");
        }
    }
  //DISPLAYING METHOD
    private class DisplayButton implements ActionListener
    {

        @Override
        public void actionPerformed(ActionEvent e)
        {
            String details = "";
            for (Child kid : kids)
            {
                details += kid.toString()+"\n";
            }
            area.setText(details.toString());
        }
    }
//CREATING CHILD CLASS
  public class Child
    {
        private String name;
        private String gender;
        public Child(String name, String gender)
        {
            this.name = name;
            this.gender = gender;
        }
        public String getName() {
            return name;
        }
```

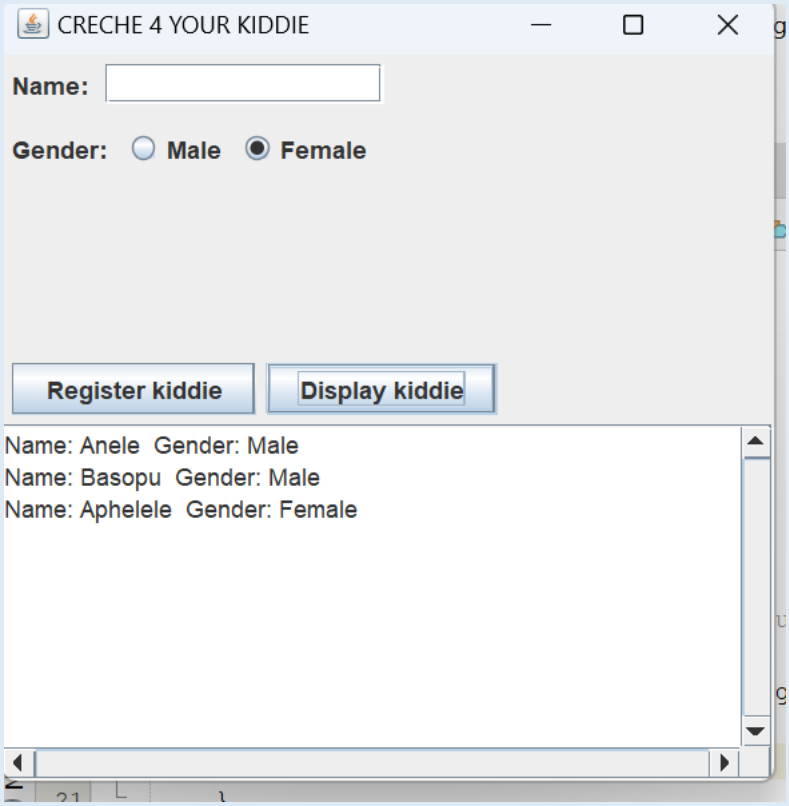
```
        public String getGender()
        {
            return gender;
        }
        @Override
        public String toString()
        {
            return "Name: " + name + "  Gender: " + gender ;
        }
    }
}

//MAIN METHOD
package crecheapp;

import za.ac.tut.gui.CreshLibrary;

/**
 *
 * @author Alex
 */
public class CrecheApp {

    public static void main(String[] args)
    {
        new CreshLibrary();
    }

}
```

- **When user did fill all the details**



- **After submitting all the details**

- **After clicking display button**


CRECHE 4 YOUR KIDDIE
Name:
Gender:
Male
Female
Register kiddie
Display kiddie
Name: Anele Gender: Male
Name: Basopu Gender: Male
Name: Aphelele Gender: Female

```
//QUESTION2
package za.ac.tut.gui;

import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.*;
import javax.swing.*;
import javax.swing.border.BevelBorder;
import javax.swing.border.LineBorder;
import javax.swing.border.TitledBorder;

/**
 *
 * @author Alex
 */
public class MessageEncryptorFrame extends JFrame
{
    private JPanel mainPnel;
    private JPanel areaPnel;
```

```
private JMenu manuName;
private JMenuBar itemHolder;

private JMenuItem openFile;
private JMenuItem encrypMessage;
private JMenuItem saveEncrMessage;
private JMenuItem clear;
private JMenuItem exit;

private JLabel heading;

private JTextArea plainMessageArea;
private JTextArea encrypedMessageArea;

private JScrollPane scroll;
private JScrollPane scrolls;

public MessageEncryptorFrame()
{
    setSize(800,500);
    setTitle("Secure Message");
```

```
        setLayout(new FlowLayout());
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //creating menu
        itemHolder = new JMenuBar();
        mainPnel = new JPanel(new FlowLayout());

        manuName = new JMenu("File");

        openFile = new JMenuItem("Open file...");
        encrypMessage = new JMenuItem("Encrypt message...");
        saveEncrMessage = new JMenuItem("Save encrypted
message...");
        clear = new JMenuItem("Clear");
        exit = new JMenuItem("Exit");

        //adding action listeners
        openFile.addActionListener(new encryptedMessages());
        encrypMessage.addActionListener(new encryptedMessages());
        saveEncrMessage.addActionListener(new encryptedMessages());
        clear.addActionListener(new encryptedMessages());
```

```
        exit.addActionListener(new encryptedMessages());


        heading = new JLabel("Message Encryptor");
        heading.setFont(new
Font(Font.SANS_SERIF,Font.ITALIC+Font.BOLD,30));
        heading.setForeground(Color.BLUE);
        heading.setBorder(new BevelBorder(BevelBorder.RAISED));


        areaPnel = new JPanel(new FlowLayout());
        //creating area
        plainMessageArea = new JTextArea(20,40);
        plainMessageArea.setBorder(new TitledBorder(new
LineBorder(Color.BLACK,2),"Plain message"));


        // added scroll pane to plainMessageArea
        scroll = new JScrollPane(plainMessageArea);

scroll.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_ALW
AYS);

scroll.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBAR
_ALWAYS);
```

```
        encrypedMessageArea = new JTextArea(20,40);
        encrypedMessageArea.setBorder(new TitledBorder(new
LineBorder(Color.BLACK,2),"Encrypted message"));

        // added scroll pane to encrypedMessageArea
        scrolls = new JScrollPane(encrypedMessageArea);

scrolls.setVerticalScrollBarPolicy(JScrollPane.VERTICAL_SCROLLBAR_AL
WAYS);

scrolls.setHorizontalScrollBarPolicy(JScrollPane.HORIZONTAL_SCROLLBA
R_ALWAYS);

        manuName.add(openFile);
        manuName.add(encrypMessage);
        manuName.add(saveEncrMessage);
        manuName.add(clear);
        manuName.add(exit);
        itemHolder.add(manuName);
        setJMenuBar(itemHolder);
```

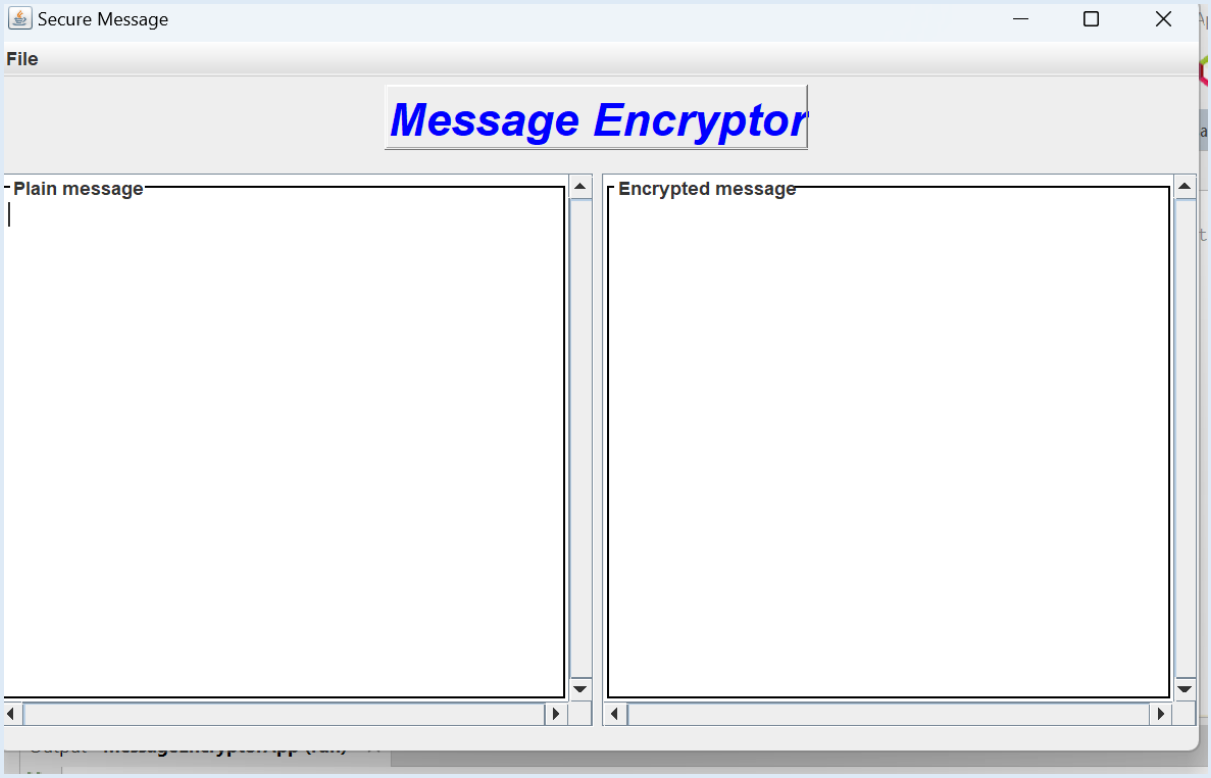
```
        areaPnel.add(scroll);
        areaPnel.add(scrolls);

        mainPnel.add(areaPnel);

        add(heading);
        add(mainPnel);
        setVisible(true);
    }

    private class encryptedMessages implements ActionListener
    {
        @Override
        public void actionPerformed(ActionEvent e)
        {
            if(e.getSource() == openFile)
            {
                JFileChooser fileSaver = new JFileChooser();
                int results =
fileSaver.showOpenDialog(MessageEncryptorFrame.this);
                if(results == JFileChooser.APPROVE_OPTION)
```

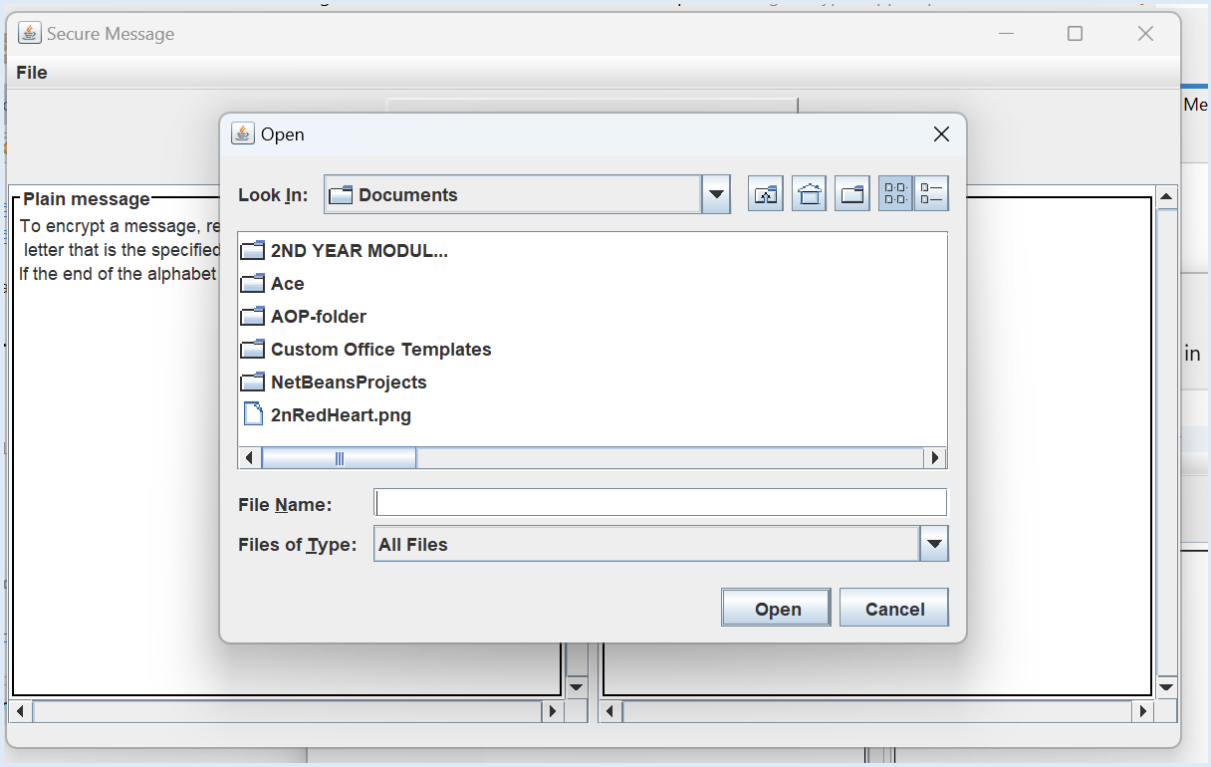
```
            {
                File file = fileSaver.getSelectedFile();
                try (BufferedReader read = new BufferedReader(new
FileReader(file)))
                {
                    plainMessageArea.read(read, "Plain message");
                }
                catch (IOException ex)
                {

JOptionPane.showMessageDialog(MessageEncryptorFrame.this,
ex.getMessage());
                }
            }
        }
        else if(e.getSource() == encrypMessage)
        {
            String plainMessage = plainMessageArea.getText();
            if(!plainMessage.isEmpty())
            {
                String encryptedMessage =
WeDoSecureApp(plainMessage, 3);
```

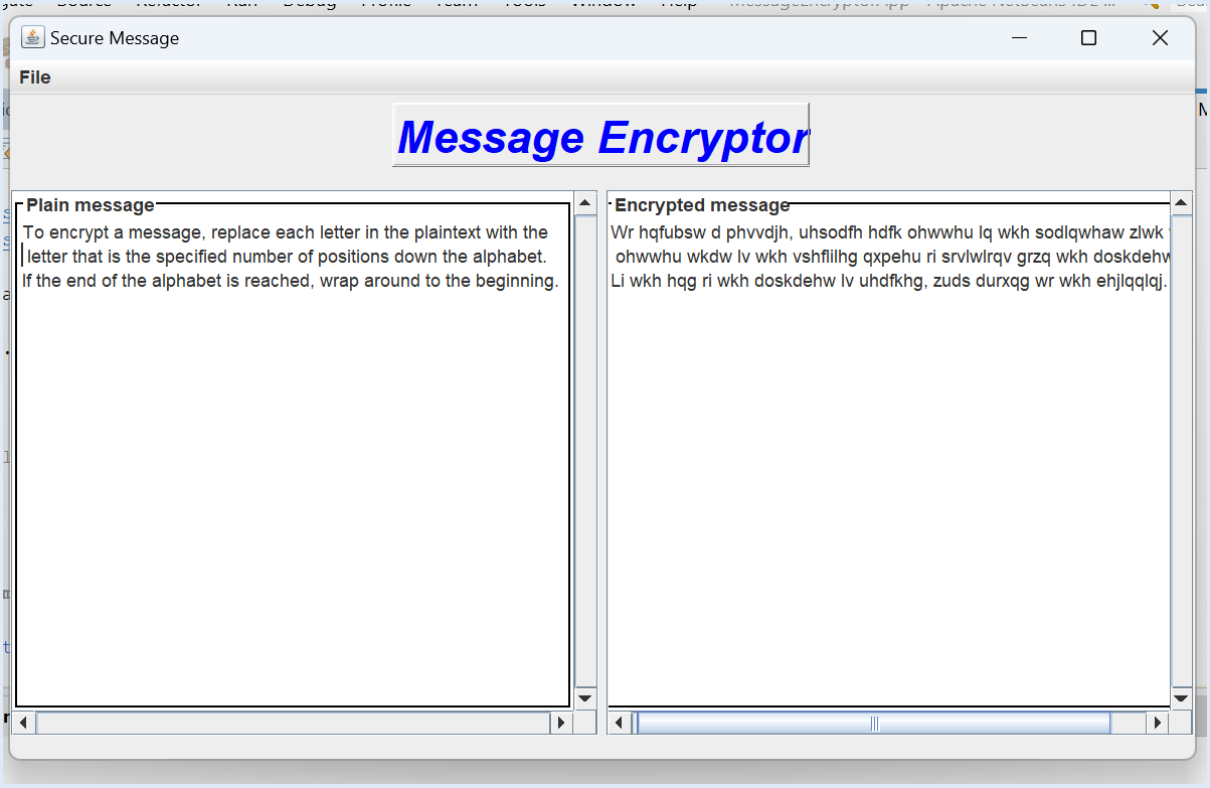
```
                encrypedMessageArea.setText(encryptedMessage);
            }
        }
        else if (e.getSource() == saveEncrMessage)
        {
            JFileChooser fileChooser = new JFileChooser();
            int result =
fileChooser.showSaveDialog(MessageEncryptorFrame.this);
            if (result == JFileChooser.APPROVE_OPTION)
            {
                File file = fileChooser.getSelectedFile();
                try (FileWriter writer = new FileWriter(file))
                {
                    encrypedMessageArea.write(writer);
                }
                catch (IOException ex)
                {

JOptionPane.showMessageDialog(MessageEncryptorFrame.this,
ex.getMessage());
                }
            }
```

```
        }
        else if (e.getSource() == clear)
        {
            plainMessageArea.setText("");
            encrypedMessageArea.setText("");
        }
        else if (e.getSource() == exit)
        {
            System.exit(0);
        }
    }
}

private String WeDoSecureApp(String message, int shift)
{
    StringBuilder encryptedText = new StringBuilder();
    shift = shift % 26 + 26;
    for (char i : message.toCharArray())
    {
        if(Character.isLetter(i))
        {
```

```
            if(Character.isUpperCase(i))
            {
                encryptedText.append((char)('A'+(i-'A'+shift)%26));
            }
            else
            {
                encryptedText.append((char)('a'+(i-'a'+shift)%26));
            }
        }
        else
        {
            encryptedText.append(i);
        }
    }
    return encryptedText.toString();
  }
}
```


Secure Message
File
Message Encryptor
Plain message
Encrypted message


Secure Message
File
Plain message
To encrypt a message, re
letter that is the specified
If the end of the alphabet
Open
Look In:
Documents
2ND YEAR MODUL...
Ace
AOP-folder
Custom Office Templates
NetBeansProjects
2nRedHeart.png
File Name:
Files of Type:
All Files
Open
Cancel


Secure Message
File
Message Encryptor
Plain message
To encrypt a message, replace each letter in the plaintext with the
letter that is the specified number of positions down the alphabet.
If the end of the alphabet is reached, wrap around to the beginning.
Encrypted message
Wr hqfubsw d phvvdjh, uhsodfh hdfk ohwwhu lq wkh sodlqwhaw zlwk
ohwwhu wkdw lv wkh vshflilhg qxpehu ri srvlwlrqv grzq wkh doskdehw
Li wkh hqg ri wkh doskdehw lv uhdfkhg, zuds durxqg wr wkh ehjlqqlqj.