Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра ЕОМ

Звіт

з лабораторної роботи № 3

з дисципліни «Захист інформації в комп’ютерних системах»

на тему: «Криптоаналіз шифрів моноалфавітної заміни»

Виконав: ст. гр. КІ-302

Радевич-Винницький Я.А.

Перевірив:

Муляревич О.В.

Мета роботи: ознайомитись з основними методами, що використовуються для криптоаналізу шифрів моноалфавітної заміни та, зокрема, з основами частотного аналізу шифрованого тексту.

Завдання:

Створити програму, що реалізує метод крипто аналізу на основі частотного аналізу шифрованого тексту

Варіант: 22

Виконання завдання:

Для виконання завдання було вибрано мову Java та бібліотеку Swing та створення графічного інтерфейсу додатку.

Програма – Frequency Analysis Decryptor

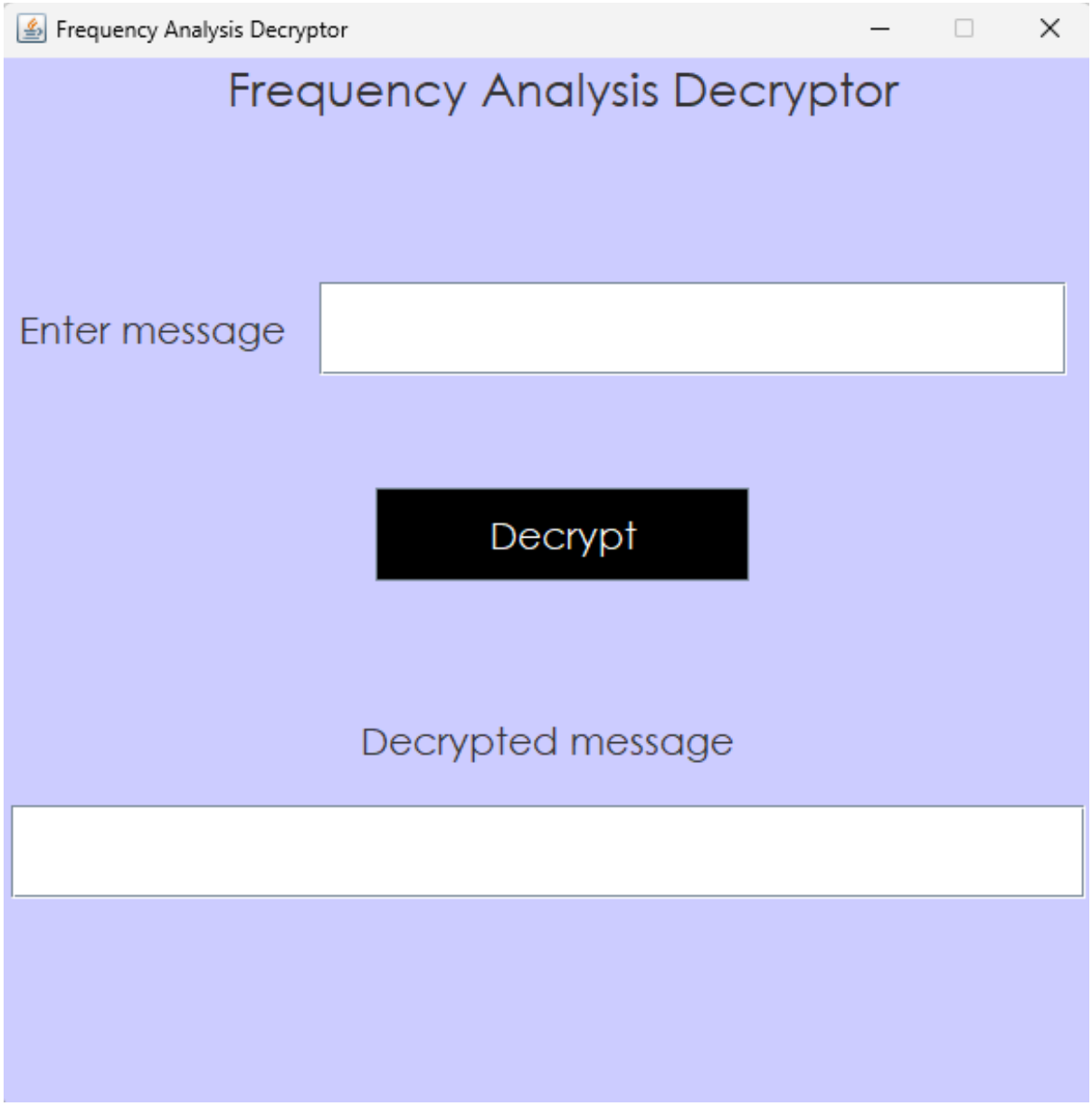

Рис. 1 – вікно програми

Код файлу *FrequencyAnalysisDecryptor.java*, у якому міститься реалізація методів криптоаналізу:

Лістинг 1

```
package decryptor;

import java.util.HashMap;
import java.util.Map;

public class FrequencyAnalysisDecryptor {
    private static final int ALPHABET_LENGTH = 26;

    private Map<Character, Double> getLettersFrequency(String text)
{
        Map<Character, Double> freq = new HashMap<>();
        int total = 0;
        for (char c : text.toCharArray()) {
            if (Character.isLetter(c)) {
                c = Character.toLowerCase(c);
                freq.put(c, freq.getOrDefault(c, 0.0) + 1);
                total++;
            }
        }
        if (total == 0) return freq;
        for (char c : freq.keySet()) {
            freq.put(c, freq.get(c) / total);
        }
        return freq;
    }

    private String decryptCaesarCipher(String ciphertext, int
offset) {
        StringBuilder decryptedText = new StringBuilder();
        for (char c : ciphertext.toCharArray()) {
            if (Character.isLetter(c)) {
                char base = Character.isUpperCase(c) ? 'A' : 'a';
                decryptedText.append((char) (((c - base - offset +
ALPHABET_LENGTH)
                        % ALPHABET_LENGTH) + base));
            } else {
                decryptedText.append(c);
            }
        }
        return decryptedText.toString();
    }

    public String frequencyAnalysis(String ciphertext) {
        Map<Character, Double> charactersFrequencyMap =
getLettersFrequency(ciphertext);
        char mostCommonLetter = ' ';
        double maxFreq = -1;
        for (char c : charactersFrequencyMap.keySet()) {
            if (charactersFrequencyMap.get(c) > maxFreq) {
                maxFreq = charactersFrequencyMap.get(c);
```

```
                mostCommonLetter = c;
            }
        }
        int offset = (mostCommonLetter - 'e' + ALPHABET_LENGTH) %
ALPHABET_LENGTH;
        return decryptCaesarCipher(ciphertext, offset);
    }
}
```

Код файлу *Frame.java*, у якому міститься код графічного інтерфейсу програми:

Лістинг 2

```
package gui;

import decryptor.FrequencyAnalysisDecryptor;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Frame extends JFrame implements ActionListener {
    private static final String FRAME_TITLE = "Frequency Analysis
Decryptor";
    private static final int DIMENSION = 600;

    private FrequencyAnalysisDecryptor frequencyAnalysisDecryptor;
    private JFrame frame;
    private JLabel headLabel;
    private JLabel inputMessageLabel;
    private JLabel outputMessageLabel;
    private JTextField inputTextField;
    private JTextField outputTextField;
    private JButton decryptionButton;

    public Frame() {
        headLabel = new JLabel();
        adjustHeadLabelSettings(headLabel);

        inputMessageLabel = new JLabel();
        adjustInputMessageLabelSettings(inputMessageLabel);

        inputTextField = new JTextField();
        adjustInputTextFieldSettings(inputTextField);

        decryptionButton = new JButton();
        adjustDecryptButtonSettings(decryptionButton);

        outputMessageLabel = new JLabel();
        adjustOutputMessageLabelSettings(outputMessageLabel);

        outputTextField = new JTextField();
        adjustOutputTextFieldSettings(outputTextField);
```

```
        frame = new JFrame();
        adjustFrameSettings(frame);

        frame.add(headLabel);
        frame.add(inputMessageLabel);
        frame.add(inputTextField);
        frame.add(decryptionButton);
        frame.add(outputMessageLabel);
        frame.add(outputTextField);
    }

    private void adjustFrameSettings(JFrame frame) {
        frame.setTitle(FRAME_TITLE);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(false);
        frame.setSize(DIMENSION,DIMENSION);
        frame.getContentPane().setBackground(new
Color(204,204,255));
        frame.setLayout(null);
        frame.setVisible(true);
    }

    private void adjustHeadLabelSettings(JLabel headLabel) {
        headLabel.setText(FRAME_TITLE);
        headLabel.setFont(new Font("Century Gothic", Font.PLAIN,
25));
        headLabel.setVerticalAlignment(JLabel.TOP);
        headLabel.setHorizontalAlignment(JLabel.CENTER);
        headLabel.setBounds(0,0, DIMENSION,50);
    }

    private void adjustInputMessageLabelSettings(JLabel
inputMessageLabel) {
        inputMessageLabel.setText("Enter message");
        inputMessageLabel.setFont(new Font("Century Gothic",
Font.PLAIN, 20));
        inputMessageLabel.setHorizontalAlignment(JLabel.CENTER);
        inputMessageLabel.setBounds(5, 120, 150, 50);
    }

    private void adjustInputTextFieldSettings(JTextField
inputTextField) {
        inputTextField.setPreferredSize(new Dimension(250,40));
        inputTextField.setBounds(170, 120, 400, 50);
        inputTextField.setFont(new Font("Century Gothic",
Font.PLAIN, 20));
        inputTextField.setHorizontalAlignment(JLabel.CENTER);
    }

    private void adjustDecryptButtonSettings(JButton
decryptionButton) {
        decryptionButton.setBounds(200, 230, 200, 50);
        decryptionButton.setText("Decrypt");
        decryptionButton.setFont(new Font("Century Gothic",
```

```
Font.PLAIN, 20));
        decryptionButton.setForeground(Color.WHITE);
        decryptionButton.setFocusable(false);
        decryptionButton.setBackground(Color.black);
        decryptionButton.addActionListener(this);
    }

    private void adjustOutputMessageLabelSettings(JLabel
outputMessageLabel) {
        outputMessageLabel.setText("Decrypted message");
        outputMessageLabel.setFont(new Font("Century Gothic",
Font.PLAIN, 20));
        outputMessageLabel.setHorizontalAlignment(JLabel.CENTER);
        outputMessageLabel.setBounds(5, 340, 575, 50);
    }

    private void adjustOutputTextFieldSettings(JTextField
outputTextField) {
        outputTextField.setPreferredSize(new Dimension(575,50));
        outputTextField.setBounds(5, 400, 575, 50);
        outputTextField.setFont(new Font("Century Gothic",
Font.PLAIN, 20));
        outputTextField.setHorizontalAlignment(JLabel.CENTER);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        if (e.getSource().equals(decryptionButton)) {
            frequencyAnalysisDecryptor = new
FrequencyAnalysisDecryptor();
            String decryptedMessage =
frequencyAnalysisDecryptor.frequencyAnalysis(inputTextField.getText
());
            outputTextField.setText(decryptedMessage);
        }
    }
}
```

Код головного файлу програми - Main.java:

Лістинг 3

```
import gui.Frame;

public class Main {
    public static void main(String[] args) {
        Frame frame = new Frame();
    }
}
```

Результат роботи програми:

Зашифроване повідомлення: uf ime pueoxaeqp kqefqdpmk ftmf eqhqdmx uzradymx ngf pudqof oazfmofe tmhq nqqz ympq iuft baxufuomx dqbdqeqzfmfuhqe az ftq huqf oazs uz yaeoai

Розшифроване повідомлення: it was disclosed yesterday that several informal but direct contacts have been made with political representatives on the viet cong in moscow

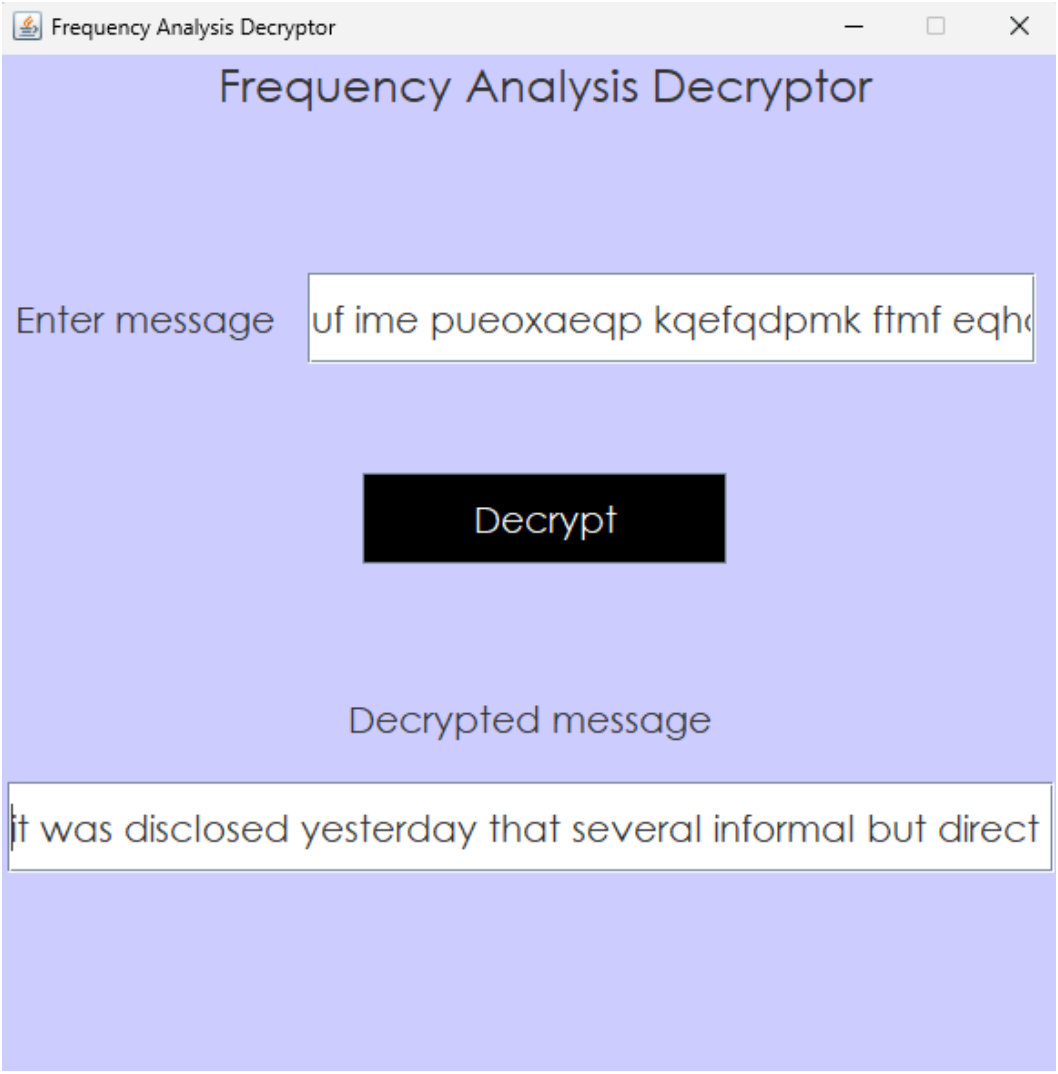

Рис. 2 – результат роботи програми

Висновок: у ході виконання лабораторної роботи було вивчено основні методи криптоаналізу шифрів моноалфавітної заміни та основи частотного аналізу шифрованого тесту. Було створено програму, що реалізує метод криптоаналізу на основі частотного аналізу шифрованого тексту.