МІНІСТЕРСТВО ОБОРОНИ УКРАЇНИ

ВІЙСЬКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ ТА ІНФОРМАТИЗАЦІЇ

КАФЕДРА КІБЕРЗБЕЗПЕКИ

**ЗВІТ**

про виконання практичної роботи:

з дисципліни «Прикладна криптологія»

Дата 09.03.2024

**Виконав:** курсант 314 навчальної групи

старший солдат\_\_\_\_\_\_\_\_\_\_\_КОНОНУЧЕНКО А.В.

**Перевiрив:** старший викладач кафедри 33

капітан\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ПРИЙМА О.О.

Завдання 1.1

ПРОГРАМНИЙ КОД String.java

import java.util.Scanner;

public class String {

public static java.lang.String filterText(java.lang.String text, java.lang.String alphabet, java.lang.String mode) {

if (mode.equals("normal")) {

alphabet += alphabet.toUpperCase() + "abcdefghijklmnopqrstuvwxyz";

} else if (mode.equals("with\_space")) {

alphabet += alphabet.toUpperCase() + "abcdefghijklmnopqrstuvwxyz";

} else {

throw new IllegalArgumentException("Invalid mode. Please choose 'normal' or 'with\_space'.");

}

if (mode.equals("normal")) {

// Заміна всіх символів, крім літер з алфавіту, на пробіли

java.lang.String filteredText = text.replaceAll("[^" + alphabet + "]", "");

// Заміна прописних літер на строчні

filteredText = filteredText.toLowerCase();

return filteredText;

} else if (mode.equals("with\_space")) {

// Заміна всіх символів, крім літер з алфавіту, на пробіли

java.lang.String filteredText = text.replaceAll("[^" + alphabet + "]", " ");

// Заміна послідовності з декількох пробілів на один пробіл

filteredText = filteredText.replaceAll("\\s+", " ");

// Видалення пробілів на початку та в кінці тексту

filteredText = filteredText.trim();

// Заміна прописних літер на строчні

filteredText = filteredText.toLowerCase();

return filteredText;

} else {

throw new IllegalArgumentException("Invalid mode. Please choose 'normal' or 'with\_space'.");

}

}

public static void main(java.lang.String[] args) {

Scanner scanner = new Scanner(System.in);

boolean exit = false;

while (!exit) {

System.out.println("Enter the text:");

java.lang.String text = scanner.nextLine();

java.lang.String ukrainianAlphabet = "абвгґдеєжзиіїйклмнопрстуфхцчшщьюя";

java.lang.String englishAlphabet = "abcdefghijklmnopqrstuvwxyz";

System.out.println("Choose the filtering mode:");

System.out.println("1. Normal alphabet");

System.out.println("2. Alphabet with spaces");

int modeChoice = scanner.nextInt();

scanner.nextLine();

java.lang.String filteredText;

if (modeChoice == 1) {

filteredText = filterText(text, ukrainianAlphabet + englishAlphabet, "normal");

} else if (modeChoice == 2) {

filteredText = filterText(text, ukrainianAlphabet + englishAlphabet, "with\_space");

} else {

System.out.println("Invalid mode choice. Using default mode (Normal alphabet).");

filteredText = filterText(text, ukrainianAlphabet + englishAlphabet, "normal");

}

System.out.println("Filtered text:");

System.out.println(filteredText);

System.out.println("Do you want to continue? (yes/no)");

java.lang.String choice = scanner.nextLine();

if (!choice.equalsIgnoreCase("yes")) {

exit = true;

}

}

scanner.close();

}

}

Результатом програмного кода є фільтрація текст з видаленням усіх символів які не належать до алфавіту є дві операції обробки тексту:

1.Без пробілів (without spaces)

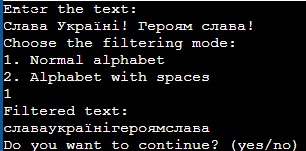


рис.1

2.З пробілами (with spaces)

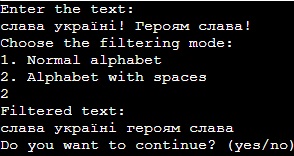


рис.2

Завдання 1.2

ПРОГРАМНИЙ КОД FrequencyAnalysisGUI.java

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.File;

import java.io.FileNotFoundException;

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

public class FrequencyAnalysisGUI {

public static void main(String[] args) {

SwingUtilities.invokeLater(() -> {

JFrame frame = new JFrame("Frequency Analysis");

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

frame.setSize(600, 400);

JTextArea inputArea = new JTextArea(10, 40);

JTextArea outputArea = new JTextArea(10, 40);

outputArea.setEditable(false);

JButton button = new JButton("Choose File");

button.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

JFileChooser fileChooser = new JFileChooser();

int returnValue = fileChooser.showOpenDialog(null);

if (returnValue == JFileChooser.APPROVE\_OPTION) {

File selectedFile = fileChooser.getSelectedFile();

performFrequencyAnalysis(selectedFile, outputArea);

}

}

});

JButton button2 = new JButton("Analyze Text");

button2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String text = inputArea.getText();

performFrequencyAnalysis(text, outputArea);

}

});

JPanel panel = new JPanel();

panel.add(button);

panel.add(button2);

panel.add(new JScrollPane(inputArea));

panel.add(new JScrollPane(outputArea));

frame.getContentPane().add(panel, BorderLayout.CENTER);

frame.setVisible(true);

});

}

private static void performFrequencyAnalysis(File file, JTextArea outputArea) {

String text = "";

try {

Scanner scanner = new Scanner(file);

while (scanner.hasNextLine()) {

text += scanner.nextLine();

}

scanner.close();

} catch (FileNotFoundException e) {

System.out.println("Файл не знайдено");

e.printStackTrace();

}

performFrequencyAnalysis(text, outputArea);

}

private static void performFrequencyAnalysis(String text, JTextArea outputArea) {

text = text.replaceAll("[^а-яА-Я]", "").toLowerCase();

Map<Character, Integer> charFrequencies = calculateCharFrequencies(text);

Map<String, Integer> bigramFrequencies = calculateBigramFrequencies(text);

StringBuilder output = new StringBuilder();

output.append("Частоти символів:\n");

for (Map.Entry<Character, Integer> entry : charFrequencies.entrySet()) {

output.append(entry.getKey() + ": " + (double) entry.getValue() / text.length() + "\n");

}

output.append("Частоти біграм:\n");

for (Map.Entry<String, Integer> entry : bigramFrequencies.entrySet()) {

output.append(entry.getKey() + ": " + (double) entry.getValue() / (text.length() - 1) + "\n");

}

outputArea.setText(output.toString());

}

private static Map<Character, Integer> calculateCharFrequencies(String text) {

Map<Character, Integer> frequencies = new HashMap<>();

for (char c : text.toCharArray()) {

frequencies.put(c, frequencies.getOrDefault(c, 0) + 1);

}

return frequencies;

}

private static Map<String, Integer> calculateBigramFrequencies(String text) {

Map<String, Integer> frequencies = new HashMap<>();

for (int i = 0; i < text.length() - 1; i++) {

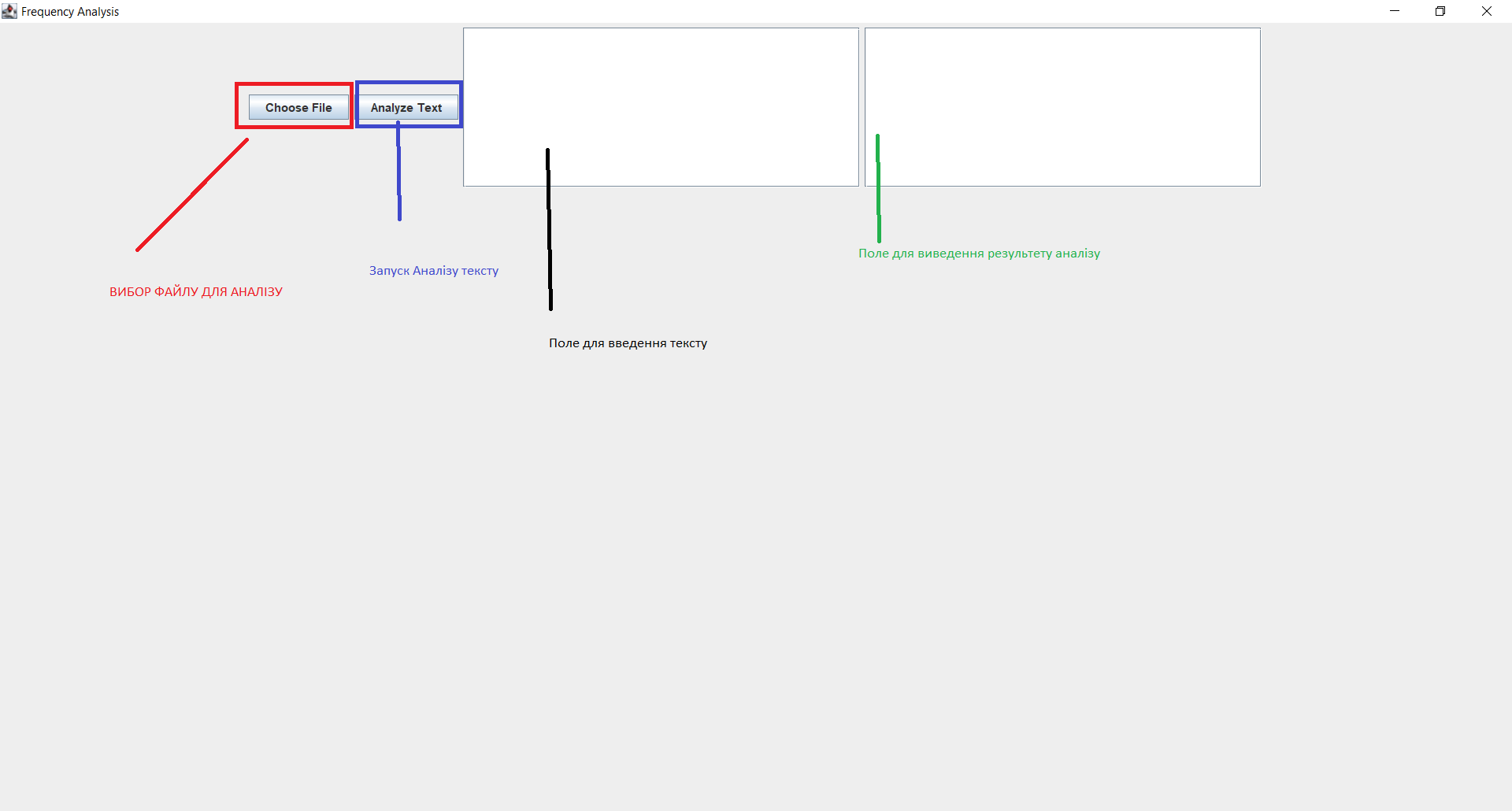
String bigram = text.substring(i, i + 2);

frequencies.put(bigram, frequencies.getOrDefault(bigram, 0) + 1);

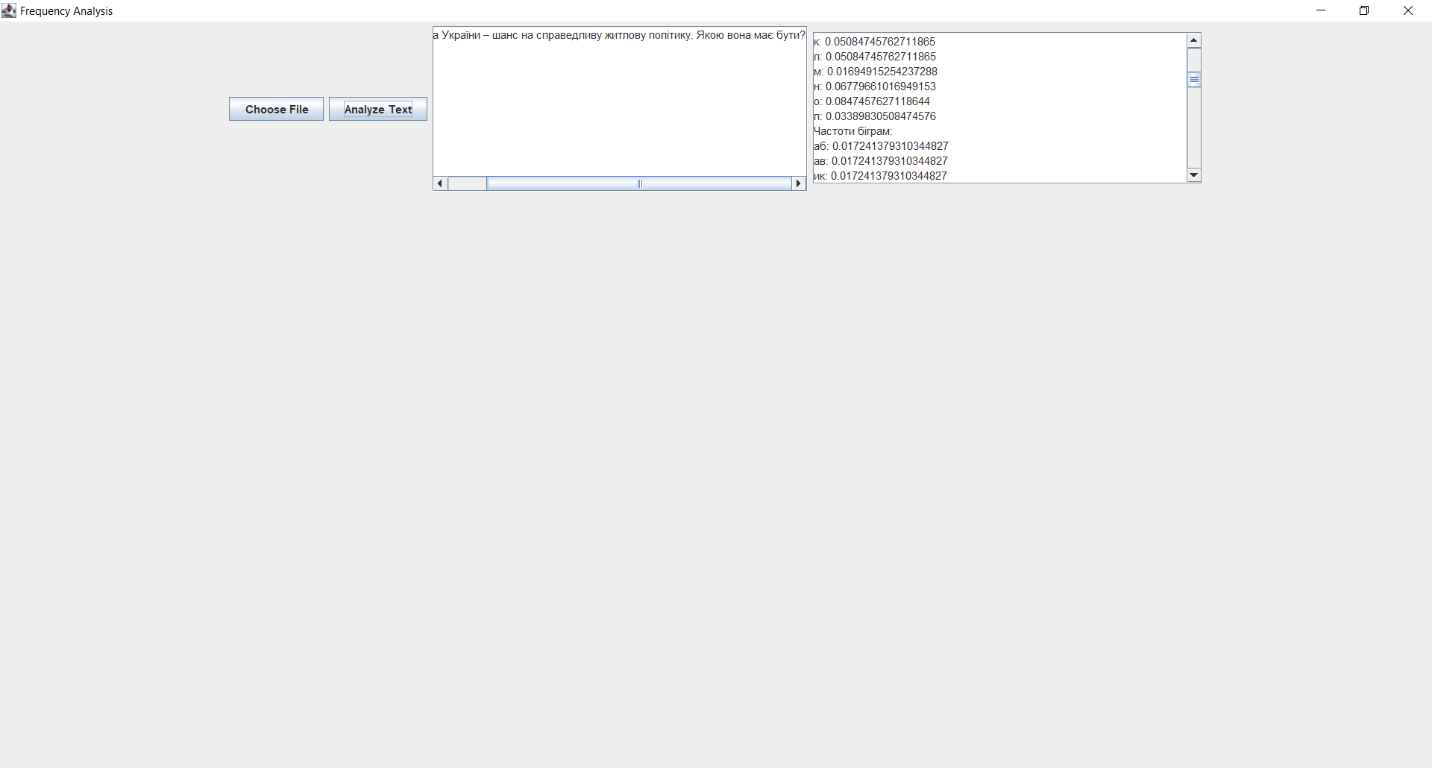
}

return frequencies;

}

 ГРАФІЧНИЙ ІНТЕРФЕЙС ПРОГРАМИ:

Після введення вибраного нами тексту або зчитання з файлу программа зчитує кожен символ та аналізує Частоти символів та Частоти біграм .



Завдання 1.3

Значення обчислених індексів відповідності

import java.util.Scanner;

public class IndexOfCoincidence {

public static void main(String[] args) {

// Запрошуємо користувача ввести текст

Scanner scanner = new Scanner(System.in);

System.out.println("Введіть текст:");

String text = scanner.nextLine();

scanner.close();

// Розрахунок індексу відповідності

double indexOfCoincidence = calculateIndexOfCoincidence(text);

// Виведення результатів

System.out.println("Індекс відповідності тексту: " + indexOfCoincidence);

}

private static double calculateIndexOfCoincidence(String text) {

int totalCharacters = text.length();

double indexOfCoincidence = 0.0;

// Розрахунок суми для формули індексу відповідності

for (char c = 'а'; c <= 'я'; c++) {

int occurrences = countOccurrences(text, c);

indexOfCoincidence += (double) (occurrences \* (occurrences - 1)) / (totalCharacters \* (totalCharacters - 1));

}

return indexOfCoincidence;

}

private static int countOccurrences(String text, char target) {

int count = 0;

for (char c : text.toCharArray()) {

if (Character.toLowerCase(c) == target) {

count++;

}

}

return count;

}

}

Порівняння з еталонними значеннями:

Індекс відповідності для зашумлених текстів: 0.0303

Індекс відповідності для російської мови: 0.055

