

**Міністерство освіти і науки України**  
**Національний технічний університет України «Київський політехнічний**  
**інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**

**Звіт**

з лабораторної роботи № 5 з дисципліни  
«Мова програмування Java»

«IO Streams»

Виконала

ЗПП-зп41 Федоренко О. Л.

Перевірила

Орленко С. П.

Київ 2025

## 1. Завдання

1. Напишіть програму, яка отримає рядок з максимальною кількістю слів із заданого файлу.
2. Зашифрувати вхідний символний потік за наступним алгоритмом: замінити кожний символ на такий, код якого більший на значення коду ключового символу (шифр Цезаря). При виконанні завдання:
  - a. Реалізувати метод, який реалізує шифрування
  - b. Реалізувати метод, який реалізує дешифрування.
  - c. Використати для розв'язання задачі потоки типу `FileOutputStream- FilterInputStream/FilterWriter-FilteredReader`.
3. Підрахувати частоту, з якою зустрічаються теги на сторінці по URL (URL на ваш вибір).
  - a. Вивести результати наступним чином: в порядку зростання для тегів в лексикографічному порядку
  - b. Вивести результати наступним чином: в порядку зростання по частоті появи тегів

## 2. Лістинг програмного коду

- Клас *MaxWordsFinder*

```
import java.io.*;

public class MaxWordsFinder {
    public static void main(String[] args) {
        String filePath = "input.txt";
        String maxWordsLine = "";
        int maxCount = 0;

        try (BufferedReader reader = new BufferedReader(new
        FileReader(filePath))) {
            String currentLine;
            while ((currentLine = reader.readLine()) != null)
            {

                String[] words =
currentLine.trim().split("\\s+");
                if (currentLine.trim().isEmpty()) continue;

                if (words.length > maxCount) {
                    maxCount = words.length;
                    maxWordsLine = currentLine;
                }
            }
        }
    }
}
```

```

        System.out.println("Рядок з максимальною
кількістю слів (" + maxCount + "):");
        System.out.println(maxWordsLine);
    } catch (IOException e) {
        System.err.println("Помилка читання файлу: " +
e.getMessage());
    }
}
}
}

```

### - Клас CaesarCipherApp

```

import java.io.*;

class CaesarWriter extends FilterWriter {
    private final int key;
    protected CaesarWriter(Writer out, int key) { super(out);
this.key = key; }
    @Override
    public void write(int c) throws IOException {
super.write(c + key); }
}

class CaesarReader extends FilterReader {
    private final int key;
    protected CaesarReader(Reader in, int key) { super(in);
this.key = key; }
    @Override
    public int read() throws IOException {
        int c = super.read();
        return (c == -1) ? -1 : (c - key);
    }
}

public class CaesarCipherApp {
    public static void main(String[] args) {
        int key = 5;
        String originalFile = "secret.txt";
        String encryptedFile = "encrypted.dat";

        try (BufferedWriter writer = new BufferedWriter(new
FileWriter(originalFile))) {
            writer.write("Hello Java I/O Streams! This is a
secret message.");
            System.out.println("Файл " + originalFile + "
успішно створено.");
        } catch (IOException e) {
            System.err.println("Не вдалося створити
початковий файл: " + e.getMessage());
            return;
        }
    }
}

```

```

        encrypt(originalFile, encryptedFile, key);

        decrypt(encryptedFile, key);
    }

    public static void encrypt(String from, String to, int
key) {
        try (Reader in = new FileReader(from);
            Writer out = new CaesarWriter(new
FileWriter(to), key)) {
            int c;
            while ((c = in.read()) != -1) {
                out.write(c);
            }
            System.out.println("Файл зашифровано у: " + to);
        } catch (IOException e) {
            System.err.println("Помилка при шифруванні: " +
e.getMessage());
        }
    }

    public static void decrypt(String from, int key) {
        File file = new File(from);
        if (!file.exists()) {
            System.err.println("Помилка: Файл для
дешифрування не знайдено: " + from);
            return;
        }

        try (Reader in = new CaesarReader(new
FileReader(from), key)) {
            System.out.print("Дешифрований вміст: ");
            int c;
            while ((c = in.read()) != -1) {
                System.out.print((char) c);
            }
            System.out.println();
        } catch (IOException e) {
            System.err.println("Помилка при дешифруванні: " +
e.getMessage());
        }
    }
}

```

- Клас *TagCounter*

```

import java.io.*;
import java.net.URI;
import java.net.URL;
import java.util.*;

```

```

import java.util.regex.*;

public class TagCounter {
    public static void main(String[] args) {
        String urlString = "https://www.google.com";
        Map<String, Integer> tagMap = new HashMap<>();

        try {
            URL url = new URI(urlString).toURL();

            try (BufferedReader reader = new
BufferedReader(new InputStreamReader(url.openStream()))) {
                String line;
                Pattern pattern = Pattern.compile("<([a-zA-
Z0-9]+)");

                while ((line = reader.readLine()) != null) {
                    Matcher matcher = pattern.matcher(line);
                    while (matcher.find()) {
                        String tag =
matcher.group(1).toLowerCase();
                        tagMap.put(tag,
tagMap.getOrDefault(tag, 0) + 1);
                    }
                }

                System.out.println("URL: " + urlString);
                System.out.println("\nТеги за алфавітом");
                tagMap.entrySet().stream()
                    .sorted(Map.Entry.comparingByKey())
                    .forEach(e -> System.out.printf("%-10s :
%d\n", e.getKey(), e.getValue()));

                System.out.println("\nТеги за частотою появи");
                tagMap.entrySet().stream()
                    .sorted(Map.Entry.comparingByValue())
                    .forEach(e -> System.out.printf("%-10s :
%d\n", e.getKey(), e.getValue()));

            } catch (Exception e) {
                System.err.println("Помилка при роботі з URL: " +
e.getMessage());
            }
        }
    }
}

```

### 3. Скріншоти виконання програми

1.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● Oleksandra_Fedorenko@EPUAKYIW0B31 java % clear
● Oleksandra_Fedorenko@EPUAKYIW0B31 java % cd /Users/Oleksandra_Fedorenko/Desktop/Java/j
  irtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessage
  ary/Application\ Support/Code/User/workspaceStorage/a32e8c5b0a2259cb9bec98b983f6d91d/re
  WordsFinder
  Рядок з максимальною кількістю слів (15):
  This is the longest line in this file because it has many words to count.
○ Oleksandra_Fedorenko@EPUAKYIW0B31 java %
```

2.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

● Oleksandra_Fedorenko@EPUAKYIW0B31 java % cd /Users/Oleksandra_Fedorenko/Desktop/Java/java
  irtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDetailsInExceptionMessages -
  ary/Application\ Support/Code/User/workspaceStorage/a32e8c5b0a2259cb9bec98b983f6d91d/redha
  sarCipherApp
  Файл secret.txt успішно створено.
  Файл зашифровано у: encrypted.dat
  Дешифрований вміст: Hello Java I/O Streams! This is a secret message.
○ Oleksandra_Fedorenko@EPUAKYIW0B31 java %
```

3

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

Oleksandra_Fedorenko@EPUAKYIW0B31 java % cd /Users/Oleksandra_Fed
  irtualMachines/jdk-21.jdk/Contents/Home/bin/java -XX:+ShowCodeDeta
  ary/Application\ Support/Code/User/workspaceStorage/a32e8c5b0a2259
  Counter

form      : 1
u         : 1
html      : 1
table     : 1
tr        : 1
nobr      : 2
img       : 2
meta      : 2
style     : 2
td        : 3
br        : 5
script    : 7
span      : 8
input     : 10
div       : 11
a         : 18
○ Oleksandra_Fedorenko@EPUAKYIW0B31 java %
```