



Міністерство освіти і науки України  
Національний технічний університет України  
“Київський політехнічний інститут імені Ігоря Сікорського”  
Факультет інформатики та обчислювальної техніки  
Кафедра інформаційні систем та технологій

**Лабораторна робота №14**  
із дисципліни *«Основи програмування»*  
**Тема:** «Потоки вводу-виводу»

Виконали:  
Студенти групи ІА-24  
Гуменюк Кирило  
Любченко Іоан  
Тильна Марія

Перевірив:  
Колеснік Валерій Миколайович

**Хід роботи:**

1. Ознайомитись з API класів та інтерфейсів для здійснення операцій вводу-виводу. Особливу увагу звернути на такі класи та інтерфейси:

- InputStream
  - FileInputStream
- OutputStream
  - FileOutputStream
- Reader
  - FileReader
- Writer
  - FileWriter
- AutoCloseable
  - Closable
- IOException

2. Виконати завдання з таблиці 2 відповідно до свого варіанту у таблиці 1.

- Кожне завдання має бути реалізовано як окремий клас.
- Кожен клас має складатись щонайменше з двох методів:
  - `public static void main(String[] args)` - точка входу.
  - Метод, що реалізує задане завдання. Метод має перевіряти аргументи та у разі їх помилковості аварійно закінчувати свою роботу шляхом викидання стандартного виключення `IllegalArgumentException` або `NullPointerException`. В разі неможливості виконання операції, метод повинен викидати `IOException` або `FileNotFoundException`. В жодному разі цей метод не повинен напяму взаємодіяти з користувачем через консоль або інший UI (ніколи не змішуйте бізнес-логіку та користувацький інтерфейс).
- Клас може містити інші допоміжні методи.
- При виконанні завдань слід звернути увагу на ефективність з точки зору швидкодії. При виконанні завдань 1-6 слід використовувати клас `BufferedReader` та `BufferedWriter`, а при 7-11 – ні в якому разі не намагаться обробляти усі байти по одному, а використовувати методи `read(byte[] b)` та `write(byte[] b)`, які працюють з масивами.

3. Відповісти на контрольні питання

**Завдання**

2	<code>String lastShortestString(String filename)</code>  Знайти у текстовому файлі найкоротший рядок. Якщо є кілька рядків, довжина яких дорівнює мінімальній, повернути останній з них.
8	<code>void cypher(String source, String destination, byte key)</code>  Створити зашифровану копію файлу, замінивши кожен з байтів сумою за модулем 2 (XOR) його значення та ключа <code>&lt;key&gt;</code> .

## Код

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
public class Task2 {
    public static void main(String[] args) {
        String filename = "D:/lab14/task2text.txt"; // шлях до тестового файлу

        try {
            String result = Task2.lastShortestString(filename);
            System.out.println("Result: " + result);
        } catch (IOException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }

    public static String lastShortestString(String filename) throws IOException {
        if (filename == null) {
            throw new NullPointerException("Filename cannot be null.");
        }

        String result = null;
        int minLength = Integer.MAX_VALUE;

        try (BufferedReader br = new BufferedReader(new FileReader(filename))) {
            String line;
            while ((line = br.readLine()) != null) {
                int length = line.length();
                if (length < minLength) {
                    result = line;
                    minLength = length;
                } else if (length == minLength) {
                    result = line;
                }
            }
        } catch (FileNotFoundException e) {
            throw new FileNotFoundException("File not found: " + filename);
        } catch (IOException e) {
            throw new IOException("Error reading file: " + filename);
        }

        if (result == null) {
            throw new IOException("File is empty: " + filename);
        }

        return result;
    }
}
```

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
public class Task8 {
    public static void main(String[] args) {
        String sourceFile = "D:/lab14/task8text.txt"; // шлях до вихідного файлу
        String destFile = "D:/lab14/task8text_encrypted.txt"; // шлях до файлу, у
        який будуть записані зашифровані дані
        byte key = 42; // ключ шифрування

        try {
            Task8.cypher(sourceFile, destFile, key);
            System.out.println("File encrypted successfully.");
        } catch (IOException e) {
            // ...
        }
    }
}
```

```

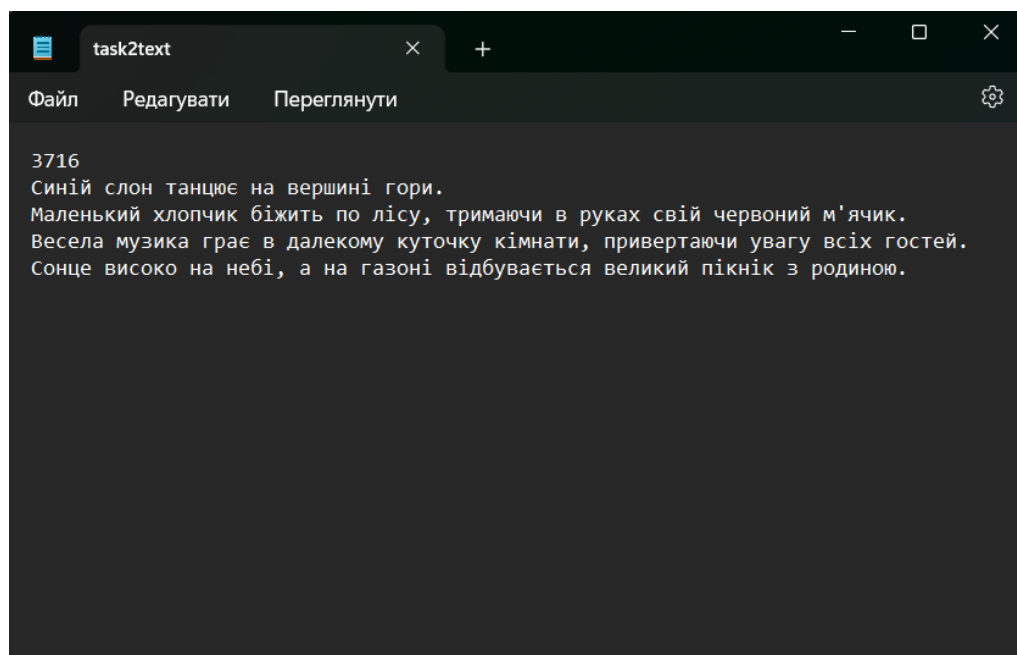
        System.err.println("Error: " + e.getMessage());
    }
}

public static void cypher(String source, String destination, byte key) throws
IOException {
    if (source == null || destination == null) {
        throw new NullPointerException("Source and destination filenames cannot be
null.");
    }

    try (FileInputStream fis = new FileInputStream(source);
        FileOutputStream fos = new FileOutputStream(destination)) {
        int b;
        while ((b = fis.read()) != -1) {
            byte encryptedByte = (byte) (b ^ key);
            fos.write(encryptedByte);
        }
    } catch (IOException e) {
        throw new IOException("Error processing files: " + e.getMessage());
    }
}
}

```

## Результат

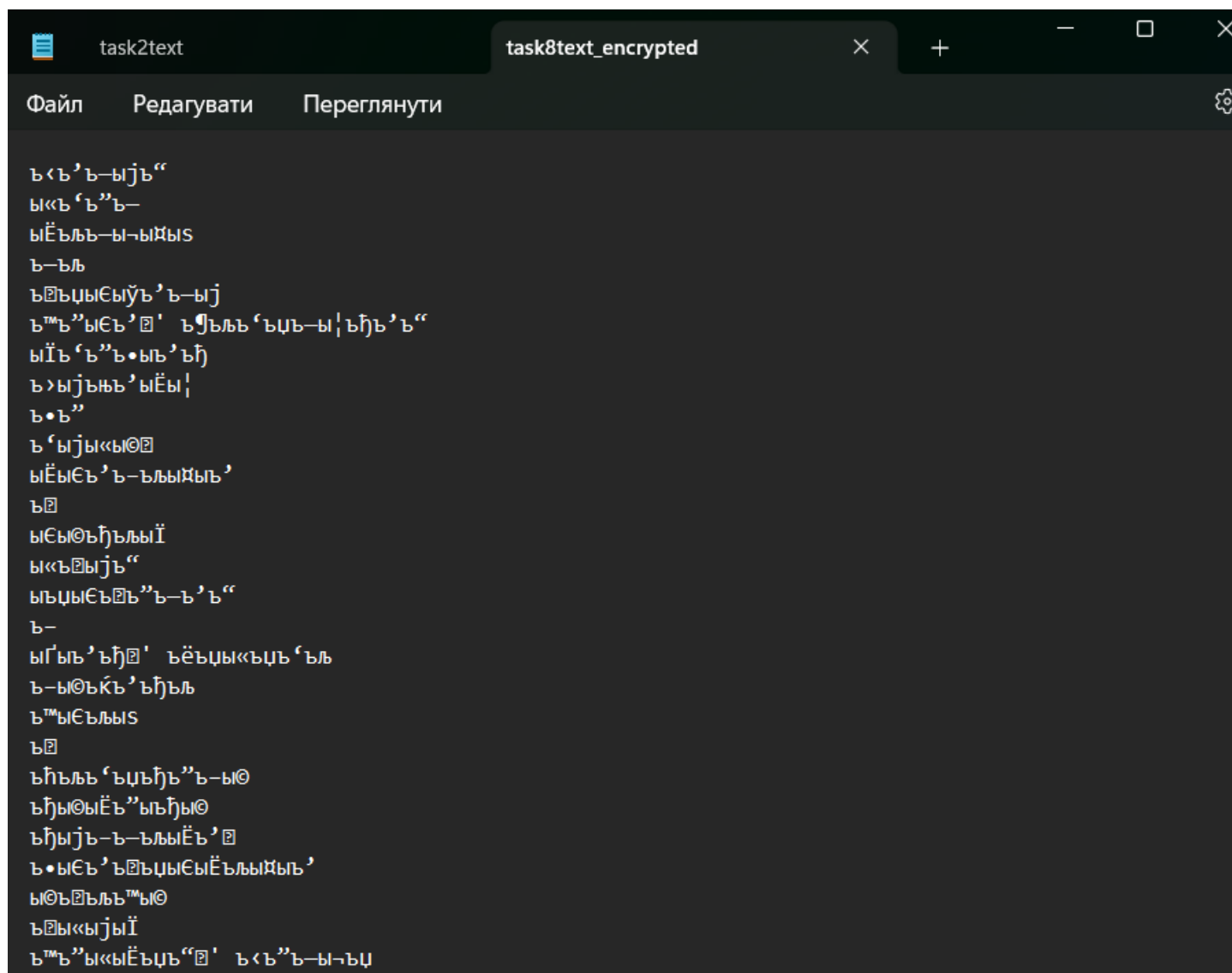


Result: 3716

Process finished with exit code 0

File encrypted successfully.

Process finished with exit code 0



task2text task8text\_encrypted

Файл Редагувати Переглянути

ь<ь'ь-ыjь"  
ы«ь'ь"ь-  
ыёьльь-ы-ыыыы  
ь-ьль  
ьёьыыёь'ь-ыj  
ь"ь"ьёь'ё' ьёьль'ьць-ы|ьёь'ь"  
ыїь'ь"ь•ыь'ьё  
ь>ыjьль'ыёь!  
ь•ь"  
ь'ьjь«ыёё  
ыёьёь'ь-ьльыь'ь  
ьё  
ыёьёьёьльыї  
ы«ьёьjь"  
ььыыёьёь"ь-ь'ь"  
ь-  
ыгь'ьёё' ьёьы«ьць'ьль  
ь-ыёькь'ьёьль  
ь"ьёьльыы  
ьё  
ьёьль'ьцьёь"ь-ыё  
ьёьёьёь"ьёьёь  
ьёьjь-ь-ьльыёь'ё  
ь•ыёь'ьёьыыёьёьльыь'ь  
ыёьёьль"ьё  
ьёь«ыjьї  
ь"ь"ь«ыёьць"ё' ь<ь"ь-ы-ьц

**Висновок:** на цій лабораторній роботі ми Ознайомились з API класами та інтерфейсами для здійснення операцій вводу-виводу. Дізнались, що таке блокуючий і неблокуючий ввід-вивід та чим InputStream відрізняється від Reader. Також закріпили ці знання на практиці