



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

Лабораторна робота №4

Потоки вводу-виводу

Виконали:
студенти групи ІА-23
Пожар Д. Ю.
Хохол М. В.
Тюска А. Ю.

Перевів:
Колеснік В. М.

Київ 2023

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 напряму 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 намагатись обробляти усі байти по одному, а використовувати методи read(byte[] b) та write(byte[] b), які працюють з масивами.

1	String firstLongestString(String filename) Знайти у текстовому файлі рядок найбільшої довжини. Якщо є кілька рядків, довжина яких дорівнює максимальній, повернути перший з них.
---	--

7	byte checkSum(String filename) Підрахувати контрольну суму бінарного файлу як суму за модулем 2 (XOR) усіх його байтів.
---	---

3. Відповісти на контрольні питання

```

import java.io.*;
import java.io.IOException;

public class Main {
    public static void main(String[] args) {
        try {
            File file = new File("test.txt");

            if (!file.exists()) file.createNewFile();

            PrintWriter pw = new PrintWriter(file);
            pw.println("OKOKOKOKOK");
            pw.println("ASSIOT");
            pw.println("ASSIOTawdad");
            pw.println("ASSIOTw");
            pw.println("ASSIOawdawdawdT");
            pw.println("ASSIOTqwdwfqwfwfwqfqw");
            pw.close();

            System.out.println(firstLongestString(file));
            System.out.println(checkSum(file));

        } catch (IOException e) {
            System.out.println("Error: " + e);
        }
    }

    public static String firstLongestString(File fileName) throws
    FileNotFoundException {
        String theLargestString = "";

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {
            String line;
            while ((line = br.readLine()) != null) {
                if (line.length() > theLargestString.length())
                    theLargestString = line;
            }
        } catch (FileNotFoundException e) {
            System.out.println("Error: " + e);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        return theLargestString;
    }

    public static byte checkSum(File fileName) throws FileNotFoundException {
        int checkSum = 0;

        try (FileInputStream input = new FileInputStream(fileName)) {
            int byteValue;
            while ((byteValue = input.read()) != -1) {
                checkSum ^= byteValue;
            }
        } catch (FileNotFoundException e) {
            System.out.println("Error: " + e);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        return (byte) checkSum;
    }
}

```

Результат роботи програми:

```
ASSIOTqwdwfqwfwfwqfqw  
115
```

Висновок

Виконавши лабораторну роботу ми навчились працювати з файлами та іншими джерелами даних за допомогою класів `FileInputStream`, `FileOutputStream`, `FileReader` та `FileWriter`. Ми також дізнались про інтерфейси `AutoCloseable` та `Closable`, які допомагають відповідно автоматично закривати ресурси та визначають метод для закриття ресурсу. Крім того, зрозуміли важливість обробки винятків `IOException` при роботі з файлами та іншими введеннями-виведеннями, щоб запобігти непередбачуваним помилкам та збоїв в програмі.