Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

# Лабораторна робота №4

з дисципліни «Основи Програмування»

**Тема:Потоки вводу-виводу**

Виконали:
Студенти групи ІА-23
Мозоль Владислав,
Курач Владислав,
Лядський Дмитро
Дата здачі:09.03.2023

Перевірив:
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[] agrs) - 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.

| | |
|---|---|
| 5 | long numberOfWords(String filename)<br><br>Підрахувати кількість слів у файлі. Під «словами» маються на увазі будь-які послідовності символів (окрім пустих), що розділені символами *<пробіл>* або *<новий рядок>*. |

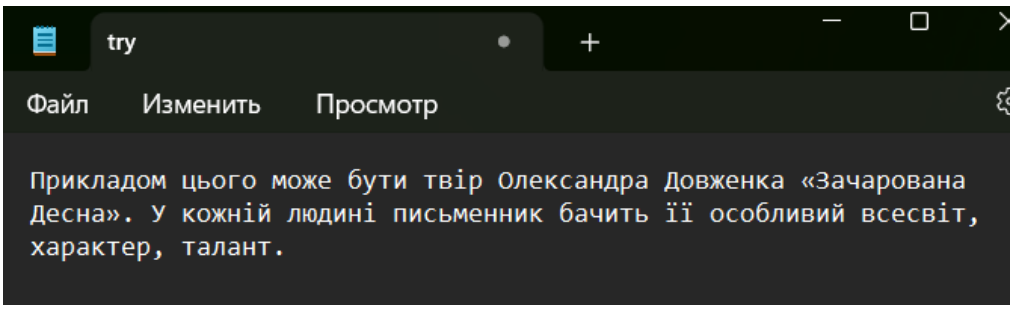
| | |
|---|---|
| 11 | void unsplit (String sourcePrefix, String destination)<br><br>Є файли з назвами *<sourcePrefix>*+".000", *<sourcePrefix>*+".001", *<sourcePrefix>*+".002", ... . Скопіювати зміст цих файлів у файл *destination*. |

```java
import java.io.*;


public class Main {
    public static void main(String[] args) {
        task5( filename: "E:\\try.txt");
        task11( sourcePrefix: "E:\\sourcePrefix000.txt",  destination: "E:\\destination.txt");
        task11( sourcePrefix: "E:\\sourcePrefix001.txt",  destination: "E:\\destination.txt");
        task11( sourcePrefix: "E:\\sourcePrefix002.txt",  destination: "E:\\destination.txt");
    }

    1 usage
    static void task5(String filename) {
        try {
            System.out.println("Кількість слів у файлі: " + numberOfWords(filename));
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        } catch (IOException e) {
            System.out.println("Some problems with IO.");
        }
    }

    3 usages
    static void task11(String sourcePrefix, String destination) {
        try {
            unsplit(sourcePrefix, destination);
        } catch (FileNotFoundException e) {
            System.out.println("File not found.");
        } catch (IOException e) {
            System.out.println("Some problems with IO.");
        }
```

```
            System.out.println("File not found.");
        } catch (IOException e) {
            System.out.println("Some problems with IO.");
        }
    }

    1 usage
    static long numberOfWords(String filename) throws IOException {
        if (filename == null) {
            throw new NullPointerException("Назва файлу не може бути пустою.");
        }
        int counter = 0;
        int checker = -1;
        FileInputStream inputStream = new FileInputStream(filename);
        BufferedInputStream bufferedInputStream = new BufferedInputStream(inputStream);
        int value;
        while ((value = bufferedInputStream.read()) != -1) {
            if ((checker != ' ' && checker != '\n') && (value == ' ' || value == '\n')) {
                counter++;
            }
            checker = value;
        }
        bufferedInputStream.close();
        inputStream.close();
        return ++counter;
    }
```

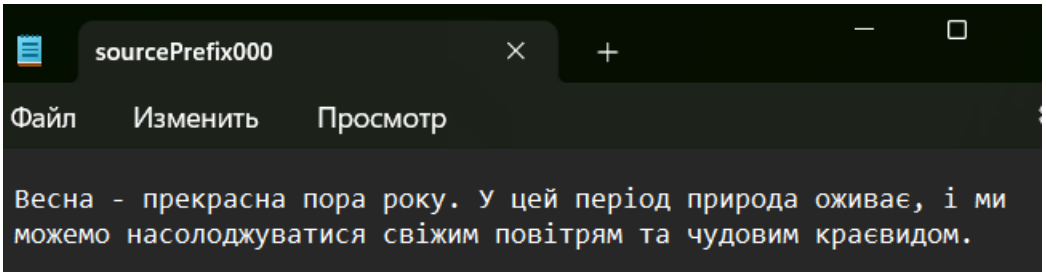
```
    1 usage
    static void unsplit(String sourcePrefix, String destination) throws IOException {
        if (sourcePrefix == null || destination == null) {
            throw new NullPointerException("Назва файлу не може бути пустою.");
        }
        char[] buf = new char[1024];
        try (Reader readerOfSource = new FileReader(sourcePrefix);
             Writer writerInDestination = new FileWriter(destination, append: true)) {

            int readed1;
            while ((readed1 = readerOfSource.read(buf)) != -1) {
                writerInDestination.write(buf, off: 0, readed1);
            }
            writerInDestination.write( str: "\n");
        }
    }
}
```

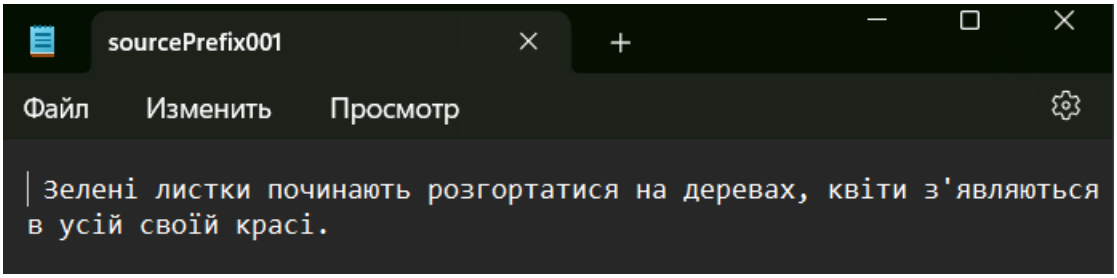
**Task 5:**

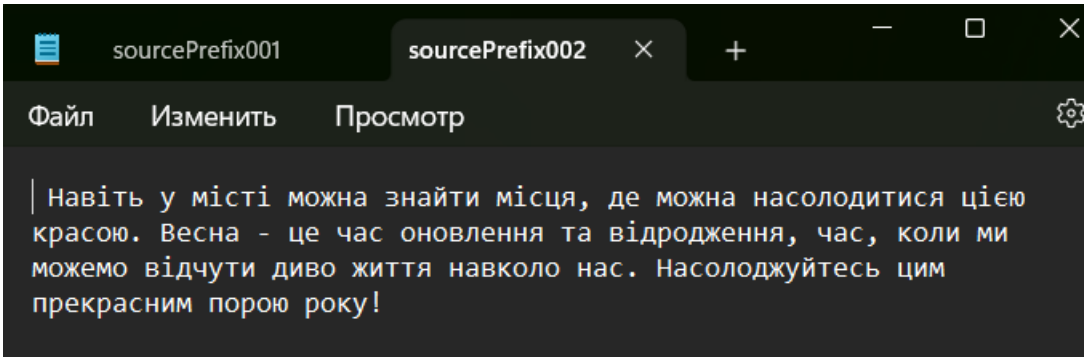


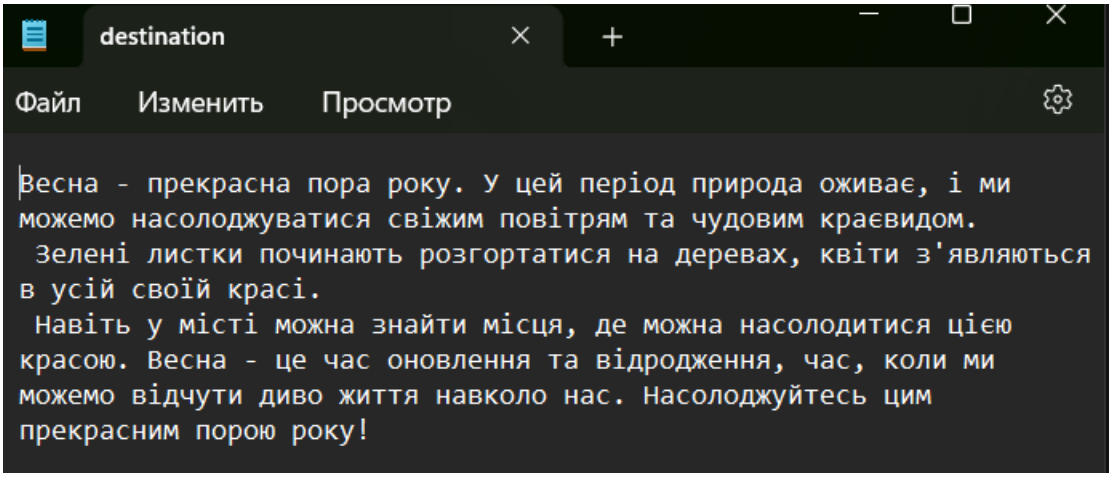
```
"F:\For programming\Java\bin\java.ex
Кількість слів у файлі: 19

Process finished with exit code 0
```

**Task 11:**




**Висновок:** виконуючи цю лабораторну роботу ми ознайомилися з API класів та інтерфейсів для здійснення операцій вводу-виводу.