



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 №3
із дисципліни «Структури даних та алгоритми»
Тема: «Потоки вводу-виводу»

Виконали:
Студенти групи ІА-24
Сіденко Д.Д.
Філімонов Є.А.
Іскандаров Е. Е. огли
Яблонський Д.Б.

Перевірив:
Колеснік Валерій Микола

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

3. Відповісти на контрольні питання

	<code>String lastShortestString(String filename)</code>
2	Знайти у текстовому файлі найкоротший рядок. Якщо є кілька рядків, довжина яких дорівнює мінімальній, повернути останній з них.

Лістинг програми:

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;

public class ShortestString {
    public static void main(String[] args) {
        catchMist();
    }
    static void catchMist() {
        try {
            String filename = "task2lab3.txt";
            System.out.println(lastShortestString(filename));
        } catch (IOException e) {
            System.out.println("Ooops, IO exception!");
        }
    }
}
```

```

        static String lastShortestString(String filename) throws IOException {
            String result="";
            try (BufferedReader reader = new BufferedReader(new
FileReader(filename))) {
                String line;
                result = reader.readLine();
                while ((line = reader.readLine()) != null) {
                    if (line==null || line.isEmpty()){
                        continue;
                    }
                    else if(line.length()<=result.length()){
                        result=line;
                    }
                }
            } catch (FileNotFoundException e) {
                throw new FileNotFoundException("File not found: " + filename);
            }
            return result;
        }
    }
}

```

	long longestZeroSequenceLength(String filename)
9	Знайти довжину найдовшої послідовності двійкового файлу, що складається з байтів зі значенням 0.

```

import java.io.*;

public class ZeroSequence {
    public static void main(String[] args) {
        catchMist();
    }
    static void catchMist(){
        try {
            String filename = "binaryFile.bin";
            System.out.println(longestZeroSequenceLength(filename));
        } catch (IOException e) {
            System.out.println("Ooops, IO exception!");
        }
    }
    static long longestZeroSequenceLength(String filename) throws IOException
{
    File file = new File(filename);

    if (!file.exists() || !file.isFile()) {
        throw new FileNotFoundException("File not found.");
    }

    long maxZeroSeqLength = 0;
    long currentZeroSeqLength = 0;

    try (FileInputStream fis = new FileInputStream(file)) {
        byte[] buffer = new byte[1024];
        int bytesRead;
        while ((bytesRead = fis.read(buffer)) != -1) {
            for (int i = 0; i < bytesRead; i++) {
                byte b = buffer[i];
                if (b == 0) {
                    currentZeroSeqLength++;
                } else {
                    if (currentZeroSeqLength > maxZeroSeqLength) {
                        maxZeroSeqLength = currentZeroSeqLength;
                    }
                }
            }
        }
    }
}

```

```

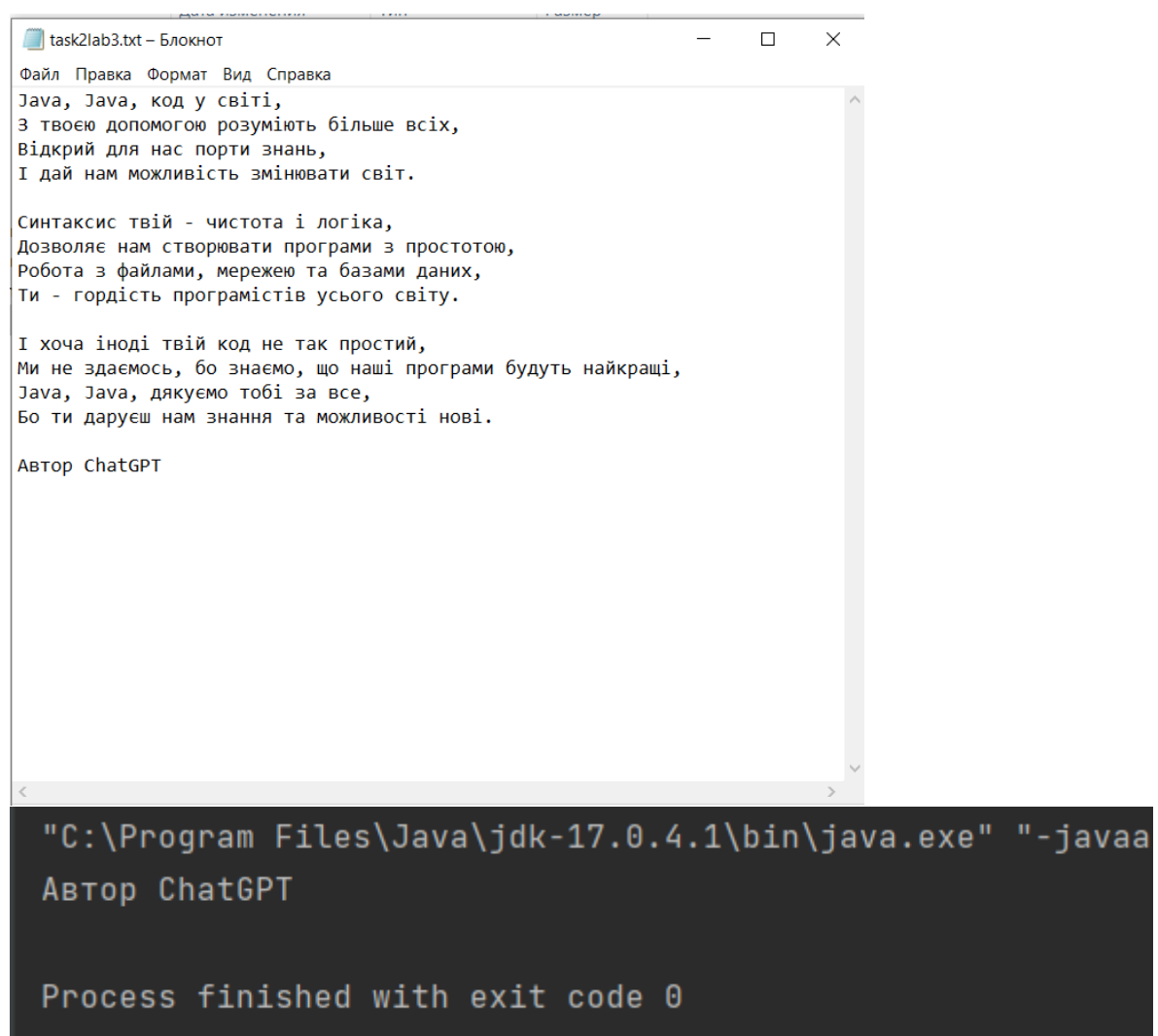
        currentZeroSeqLength = 0;
    }
}

    if (currentZeroSeqLength > maxZeroSeqLength) {
        maxZeroSeqLength = currentZeroSeqLength;
    }
} catch (FileNotFoundException e) {
    throw new FileNotFoundException("File not found: " + filename);
}

    return maxZeroSeqLength;
}
}

```

Результати виконання:



The image shows two windows. The top window is a Notepad application titled "task2lab3.txt - Блокнот". It contains a poem in Ukrainian about Java and ChatGPT, followed by the author's name "Автор ChatGPT". The bottom window is a terminal window showing the command to run the Java program and its successful execution.

```

task2lab3.txt - Блокнот
Файл Правка Формат Вид Справка
Java, Java, код у світі,
З твоєю допомогою розуміють більше всіх,
Відкрий для нас порти знань,
І дай нам можливість змінювати світ.

Синтаксис твій - чистота і логіка,
Дозволяє нам створювати програми з простотою,
Робота з файлами, мережею та базами даних,
Ти - гордість програмістів усього світу.

І хоча іноді твій код не так простий,
Ми не здаємось, бо знаємо, що наші програми будуть найкращі,
Java, Java, дякуємо тобі за все,
Бо ти даруєш нам знання та можливості нові.

Автор ChatGPT

"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaa
Автор ChatGPT

Process finished with exit code 0

```

```
public class CreateBinaryFile {  
    public static void main(String[] args) {  
        String filename = "binaryFile.bin";  
        int fileSize = 100; // Розмір файлу у байтах  
  
        try (DataOutputStream outputStream = new DataOutputStream(new FileOutputStream(filename))) {  
            byte[] data = new byte[fileSize];  
            outputStream.write(data); // записуємо масив байтів у файл  
        } catch (IOException e) {  
            System.out.println("Помилка при створенні файлу: " + e.getMessage());  
        }  
    }  
}
```

```
"C:\Program Files\Java\jdk-17.0.4.1\bin\java.exe" "-javaagent  
100
```

```
Process finished with exit code 0
```

Висновок: В даній лабораторній роботі ми ознайомились з API класів та інтерфейсів для здійснення операцій вводу-виводу.