

**Министерство образования Республики Беларусь
Учреждение образования
«Брестский государственный технический университет»
Кафедра ИИТ**

**Лабораторная работа №2
По дисциплине «СПП»
Вариант 11**

Выполнил:
Студент 3 курса
Группы ПО-9
Лебедович В.А.
Проверил:
Крощенко А.А

Брест 2024

Лабораторная работа №2

Цель работы: приобрести базовые навыки работы с файловой системой в Java.

Задание 1: Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку: 11) Напишите программу, которая считывает текст из файла и выводит все слова, содержащиеся в таком тексте, в лексикографическом порядке следования. При этом слова, встречающиеся несколько раз, должны быть выведены единожды.

Код программы:

```
import java.util.Scanner;
import java.io.File;
import java.util.Set;
import java.util.TreeSet;
import java.io.FileNotFoundException;

public class Main {
    public static void main(String[] args) {
        try {

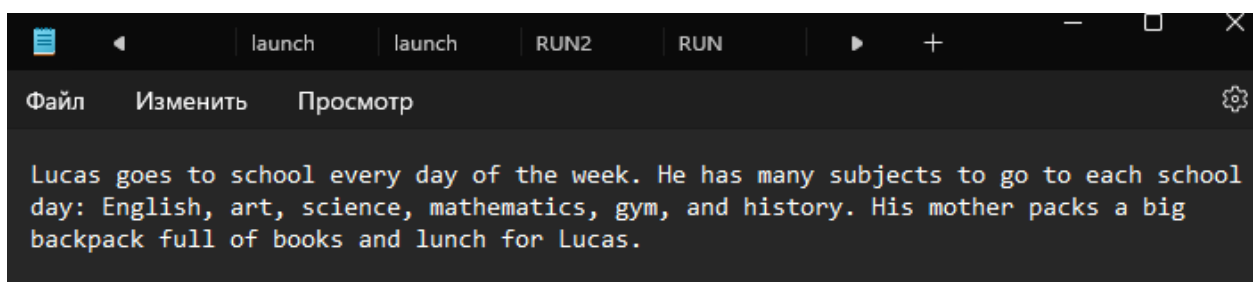
            File file = new File("input.txt");
            Scanner scanner = new Scanner(file);
            Set<String> wordsSet = new TreeSet<>();

            while (scanner.hasNext()) {
                String word = scanner.next().replaceAll("[^a-zA-Z]",
                "").toLowerCase();
                wordsSet.add(word);
            }

            for (String word : wordsSet) {
                System.out.println(word);
            }

            scanner.close();
        } catch (FileNotFoundException e) {
            System.out.println("File not found");
        }
    }
}
```

Результат работы программы:



```
PS D:\JavaProject\lab2_1\src> java Main
a
and
art
backpack
big
books
day
each
english
every
for
full
go
goes
gym
has
he
his
history
lucas
lunch
many
mathematics
mother
of
packs
school
science
subjects
the
to
week
PS D:\JavaProject\lab2_1\src> |
```

Задание 2: Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска.

Утилита tail выводит несколько (по умолчанию 10) последних строк из файла.

Формат использования: tail [-n] file

Ключ -n <количество строк> (или просто <количество строк>) позволяет изменить количество выводимых строк.

Код программы:

```
import java.io.IOException;
import java.io.RandomAccessFile;
import java.util.Iterator;
import java.util.Stack;
```

```

public class Main {
    public static void main(String[] args) {
        int count = 10; // по умолчанию выводим 10 строк
        String fileName;

        if (args.length == 4 && args[0].equals("tail") && args[1].equals("-n")) { // если ключ -n и количество строк указано через пробел
            try {
                count = Integer.parseInt(args[2]); // получаем количество
                строки

                fileName = args[3]; // получаем имя файла
            } catch (NumberFormatException e) {
                System.out.println("Invalid command");
                return;
            }
        } else if (args.length == 3 && args[0].equals("tail")) { // если
            только количество строк и имя файла указано
            try {
                count = Integer.parseInt(args[1]);
                fileName = args[2];
            } catch (NumberFormatException e) {
                System.out.println("Invalid command");
                return;
            }
        } else {
            System.out.println("Usage: tail [-n] <number of lines> file");
            return;
        }

        // читаем последние строки из файла
        try (RandomAccessFile file = new RandomAccessFile(fileName, "r")) {
            long length = file.length(); // получаем длину файла
            long pos = length - 1; // стартовая позиция - последний символ в
            файле

            int lines = 0; // количество строк
            Stack<String> tailLines = new Stack<>();

            StringBuilder builder = new StringBuilder();
            for (long pointer = pos; pointer >= 0; pointer--) {
                file.seek(pointer); // устанавливаем позицию указателя
                char c = (char) file.read(); // читаем символ
                builder.append(c); // добавляем символ в буфер

                if (c == '\n') {
                    builder = builder.reverse(); // переворачиваем буфер,
                    чтобы получить строку
                    tailLines.push(builder.toString());
                    builder.setLength(0); // очищаем буфер
                    lines++; // увеличиваем счетчик строк
                    if (lines >= count) { // если достигли нужного количества
                        строки, выходим из цикла
                        break;
                    }
                }
            }

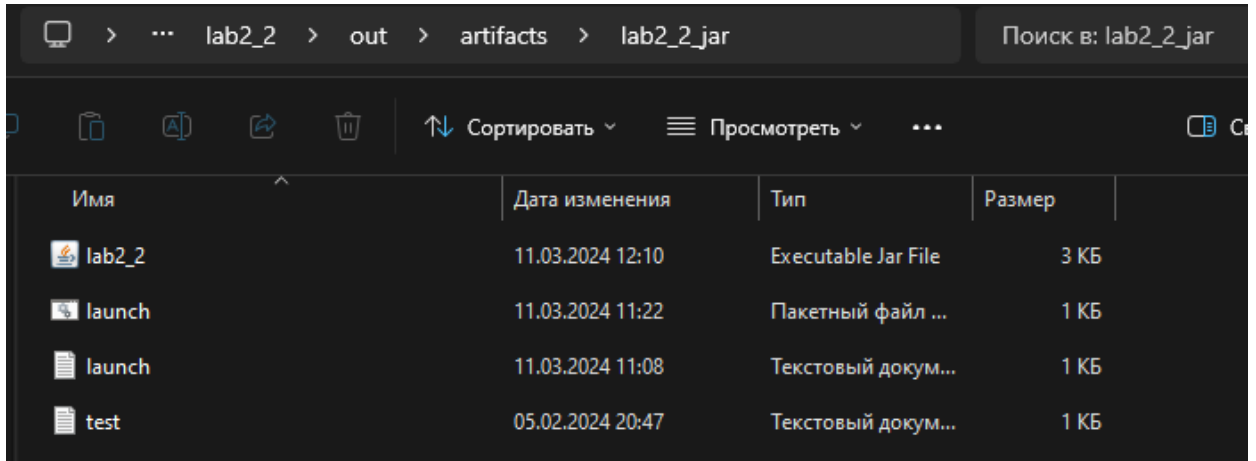
            Iterator<String> iterator = tailLines.iterator();
            while (iterator.hasNext()) {
                System.out.print(iterator.next());
            }
        } catch (IOException e) {
            System.out.println("Error reading the file: " + e.getMessage());
        }
    }
}

```

```
}  
}
```

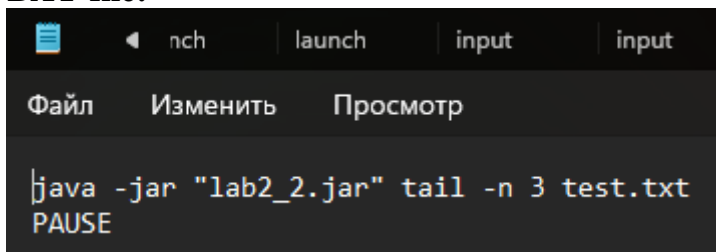
Результаты работы программы:

JAR:



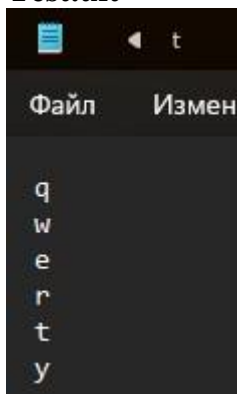
Имя	Дата изменения	Тип	Размер
lab2_2	11.03.2024 12:10	Executable Jar File	3 КБ
launch	11.03.2024 11:22	Пакетный файл ...	1 КБ
launch	11.03.2024 11:08	Текстовый докум...	1 КБ
test	05.02.2024 20:47	Текстовый докум...	1 КБ

BAT-file:

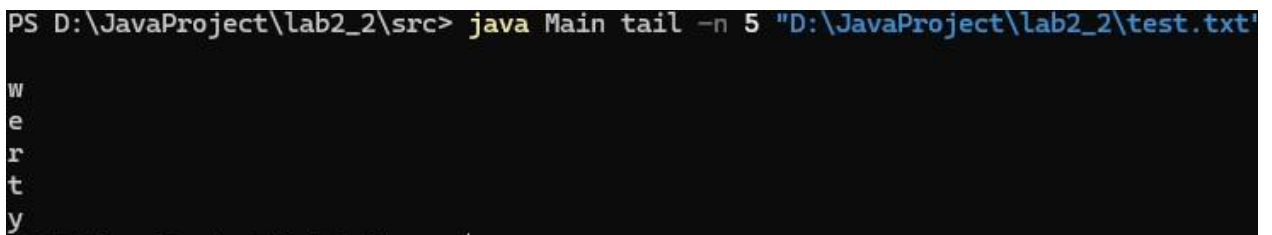


```
java -jar "lab2_2.jar" tail -n 3 test.txt  
PAUSE
```

Test.txt



```
q  
w  
e  
r  
t  
y
```



```
PS D:\JavaProject\lab2_2\src> java Main tail -n 5 "D:\JavaProject\lab2_2\test.txt"  
w  
e  
r  
t  
y
```

Вывод: приобрести базовые навыки работы с файловой системой в Java.