

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ  
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ”  
КАФЕДРА ИИТ

ОТЧЁТ  
по лабораторной работе №2

Выполнила:

студентка 3 курса  
группы ПО-9  
Шубич Дарья  
Константинова

Проверил:

Крощенко А.А.

Брест 2024

**Цель работы:**

приобрести базовые навыки работы с файловой системой в Java

**Вариант 11****Задание 1**

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.

**Входные данные:** file.txt

**Результат программы**

```
/Users/dashubaa/Library/Java/JavaVirtualMachines/openjdk-21.0.2/Content
1960s 90s a again all and around as

be block bundled can content controversy create design designs

distract doesn dry during etc experienced filler focus for

generator get history in ipsum is it laying layout

looking lorem meaningless meant natural not of on or

out own page pages paragraph passage popularity practice publishers

read s seen sentence sheets software stock surge t

text that the their to today transfer use used

very web websites when with without your
Process finished with exit code 0
```

**Код программы:**

```

public class Main {
    public static void main(String[] args) throws IOException {
        BufferedReader reader;
        try {
            reader = new BufferedReader(new FileReader("file.txt"));
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        }
        String line;
        Set<String> wordsSet = new TreeSet<>();
        while ((line = reader.readLine()) != null) {
            line = line.toLowerCase(Locale.ROOT);

wordsSet.addAll(Arrays.asList(line.split("[\\p{Punct}\\s]+")));
        }
        reader.close();
        for (String s:wordsSet){
            System.out.print(s + " ");
        }

    }
}

```

## Задание 2

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.

Пример использования:

tail -n 20 app.log

tail 20 app.log

Выводит 20 последних строк из файла app.log.

Для решения задачи подойдет класс java.io.RandomAccessFile, реализующий произвольный доступ к файлу (чтение и запись с любой позиции в файле).

**Входные данные:** file.txt

**Выходные данные:**

```
dashubaa@MacBook-Pro-Dara src % ./tail.command -n 2 file.txt
```

Use our generator to get your own, or read on for the authoritative history of lorem ipsum.%

**Код программы:**

```

public class Zad2 {
    private static final int DEFAULT_NUM_LINES = 10;

    public static void main(String[] args) {
        if (args.length < 1 || args.length > 3) {
            System.out.println("Uncorrected numbers of args. Try: tail [-n]
file");
        }
        int numLines = DEFAULT_NUM_LINES;
        String filename = null;

        if (args.length == 1){
            filename = args[0];
        } else if (args.length == 3 && args[0].equals("-n")) {
            try {
                numLines = Integer.parseInt(args[1]);
                filename = args[2];
            } catch (NumberFormatException e){
                System.out.println("Uncorrected value for numbers of
stings");
            }
        } else {
            System.out.println("Uncorrected format of args. Try: tail [-n]
file");
        }

        try(RandomAccessFile file = new RandomAccessFile(filename, "r")) {
            long fileLength = file.length();
            long position = fileLength - 1;
            int linesFound = 0;

            StringBuilder output = new StringBuilder();

            while (position >= 0 && linesFound < numLines) {
                file.seek(position);
                char c = (char) file.read();

                if (c == '\n') {
                    output.insert(0, c);
                    linesFound++;
                } else if (position == 0) {
                    output.insert(0, c);
                    linesFound++;
                } else {
                    output.insert(0, c);
                }

                position--;
            }

            System.out.print(output);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}

```