

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

Лабораторная работа №2  
по дисциплине: **СПП**  
Тема: Работа с файлами

**Выполнил**  
студент 3 курса  
Корнасеви́ч И. Д.

**Проверил**  
Кро́щенко А. А.

**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:

Напишите программу выдачи перекрестных ссылок, т.е. программу, которая выводит список всех слов документа и для каждого из этих слов печатает список номеров строк, в которые это слово входит.

#### App1.java

```
1 package main2;
2
3 import java.io.IOException;
4 import java.util.*;
5
6
7 public class App1 {
8
9     public static void main(String[] args) throws IOException {
10         var stream = App1.class.getResourceAsStream("file.txt");
11         if (stream == null) {
12             return;
13         }
14         var text = new String(stream.readAllBytes());
15         text = text.toLowerCase(Locale.ROOT);
16         var lines = text.split("\n");
17         var wordsToLines = new HashMap<String, Set<Integer>>();
18         for (var i = 0; i < lines.length; i++) {
19             var words = lines[i].split("\\W+");
20             for (var word : words) {
21                 wordsToLines.merge(word, Set.of(i), App1::merge);
22             }
23         }
24         System.out.println(wordsToLines);
25     }
26
27     private static Set<Integer> merge(Set<Integer> a, Set<Integer> b) {
28         var newSet = new HashSet<>(a);
29         newSet.addAll(b);
30         return newSet;
31     }
32 }
```

**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).

#### App2.java

```
1 package main2;
2
3 import java.io.*;
4 import java.util.Objects;
5
6
7 public class App2 {
8
9     public static void main(String[] args) {
10         var argIdx = 0;
11         var scriptFolder = args[argIdx++];
```

```
12         var lineCount = 0;
13         if (Objects.equals(args[argIdx], "-n")) {
14             argIdx++;
15         }
16         lineCount = Integer.parseInt(args[argIdx++]);
17         var file = args[argIdx];
18         var filePath = scriptFolder.concat(file);
19         try (var br = new BufferedReader(new FileReader(filePath))) {
20             String line;
21             for (var i = 0; i < lineCount && (line = br.readLine()) != null; i++) {
22                 System.out.println(line);
23             }
24         } catch (IOException e) {
25             System.out.println(e.getMessage());
26         }
27     }
28 }
```

---

## head.bat

---

```
1 @echo off
2 javac main2\App2.java
3 java main2.App2 %~dp0 %*
```

---

**Вывод** Я поработал с файлами и строками, создал скрипт для запуска java программы.