

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

Лабораторная работа №2
По дисциплине: «Современные платформы программирования»

Выполнил:
Студент 3 курса
Группы ПО-8
Шлыков А.Л.
Проверил:
Крощенко А.А.

Брест 2024

Цель работы:

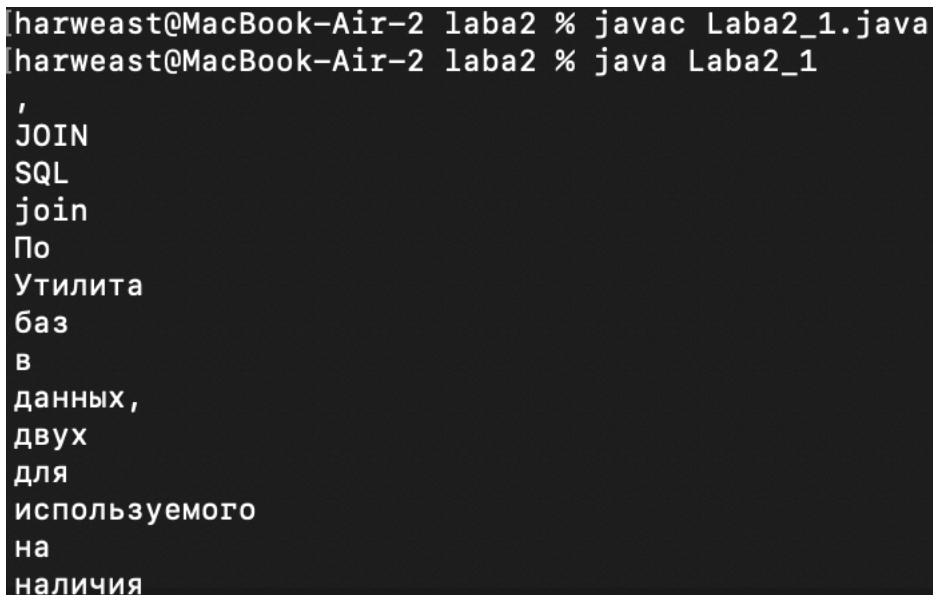
приобрести базовые навыки работы с файловой системой в Java

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.

```
public static void main(String[] args) {  
    try {  
        File file = new File("text.txt"); // Замените на путь к вашему файлу  
        BufferedReader br = new BufferedReader(new FileReader(file));  
        String st;  
        Set<String> words = new TreeSet<>();  
  
        while ((st = br.readLine()) != null) {  
            String[] lineWords = st.split("\\s+");  
            words.addAll(Arrays.asList(lineWords));  
        }  
  
        for (String word : words) {  
            System.out.println(word);  
        }  
        br.close();  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```



```
harweast@MacBook-Air-2 laba2 % javac Laba2_1.java  
harweast@MacBook-Air-2 laba2 % java Laba2_1  
,  
JOIN  
SQL  
join  
По  
Утилита  
баз  
в  
данных,  
двух  
для  
используемого  
на  
наличия
```

Задание 2

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

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).

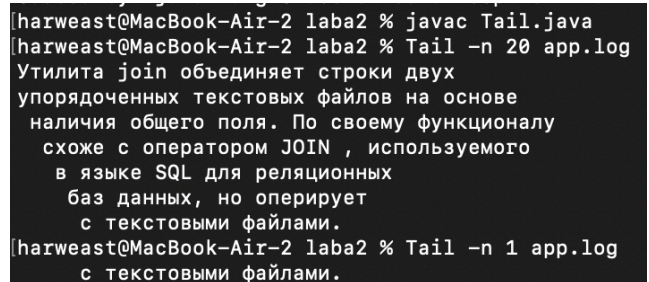
```
public static void main(String[] args) {
    int lines = 10;
    String filename;

    if (args.length == 2) {
        lines = Integer.parseInt(args[0]);
        filename = args[1];
    } else {
        filename = args[0];
    }

    try (RandomAccessFile file = new RandomAccessFile(filename, "r")) {
        long length = file.length();
        int readLines = 0;
        StringBuilder sb = new StringBuilder();

        for(long seek = length - 1; seek >= 0; seek--){
            file.seek(seek);
            int c = file.read();
            if(c == '\n'){
                readLines++;
                if(readLines == lines)
                    break;
            }
            sb.append((char)c);
        }

        System.out.println(sb.reverse().toString());
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```



```
[harweast@MacBook-Air-2 laba2 % javac Tail.java
[harweast@MacBook-Air-2 laba2 % Tail -n 20 app.log
Утилита join объединяет строки двух
упорядоченных текстовых файлов на основе
наличия общего поля. По своему функционалу
схоже с оператором JOIN , используемого
в языке SQL для реляционных
баз данных, но оперирует
с текстовыми файлами.
[harweast@MacBook-Air-2 laba2 % Tail -n 1 app.log
с текстовыми файлами.
```

Вывод: Приобрёл базовые навыки работы с файловой системой в Java

