

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

Лабораторная работа №2
По дисциплине «СПП» за 5 семестр

Выполнил:

Студент группы ПО-3

Ковалёва А. И.

Проверил:

Крощенко А. А.

Брест 2020

Вариант 12

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

Задание 1:

Напишите программу, которая случайным образом переставляет буквы в каждом слове считываемого текста и выводит преобразованный текст на экран.

Текст программы:

```
package com.company;

import java.io.*;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.*;
import java.util.stream.Collectors;

public class Main {

    static String PATH = "/Users/anastasiakovaleva/Desktop/lab-2/src/com/company/file";

    public static void main(String[] args) throws IOException {

        System.out.println(shuffle(getWords(PATH)));

    }

    public static List<String[]> getWords(String path) throws
    IOException {
        return Files
            .lines(Paths.get(path), StandardCharsets.UTF_8)
            .map(x -> x.replaceAll("[!:,.?!'\"' (+) -]", ""))
            .map(x -> x.split("\\s+"))
            .collect(Collectors.toList());
    }

    public static String shuffle(List<String []> list){
        StringBuilder stringBuilder = new StringBuilder();

        for (String[] arrStr : list) {
            for (String str : arrStr) {
                stringBuilder.append(shuffleString(str)).append(" ");
            }
        }
        return stringBuilder.toString();
    }

    public static String shuffleString(String string) {
        List<String> letters = Arrays.asList(string.split(""));
        Collections.shuffle(letters);
```

```

        StringBuilder builder = new StringBuilder();
        for (String letter : letters) {
            builder.append(letter);
        }
        return builder.toString();
    }
}

```

Результат выполнения:

noDima yraLe eiBsunss ligco is het romistnen tpra of eth ononi itohwtu
edidpecnsene to reoth sarley ti is yllttoa ilsotade It onsiact
seteunBsiEnstiis elodMs Ues Cessa and osrRtyeoip ceratsfenI shiT ayerl
duocl be taltilopnye sereud nhiwit fdenrfeit terjsocp Schu tnaoisrape
solawl fro ont usgni teh tosh app winhti eht etts atrget aubcsee on
necseineedpd aslo d3r ptyra are ddenee – sthi eamks het imnaoD Use
Cesas tests tkae utsj a wfe sdcstone tNeo Dmaoni yaerL hosdul otn
cluiend gnanhity form rehto leyras eg tProteneians – IiKUt ro fwitSUI
ro aaDt yaeLr – gMpipna lbaeCod

Задание 2:

Утилита `head` выводит несколько (по умолчанию 10) первых строк из файла. Формат использования: `head [-n] file`

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

Пример использования:

```
head -n 20 app.log
```

```
head 20 app.log
```

Выводит 20 первых 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).

Текст программы:

```

package com.company;

import java.io.File;
import java.io.IOException;
import java.io.RandomAccessFile;
import java.nio.charset.StandardCharsets;

public class Main {

    public static void main(String[] args) {
        final String dir = System.getProperty("user.dir");
        int lines = 0;
        int size = args.length;

        switch (size) {
            case 2:

```

```

        try {
            lines = Integer.parseInt(args[0]);
            String PATH = dir + "/" + args[1];
            System.out.println(headFile(new File(PATH),
lines));
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        break;
    case 3:
        try {
            lines = Integer.parseInt(args[1]);
            String PATH = dir + "/" + args[2];
            System.out.println(headFile(new File(PATH),
lines));
        } catch (Exception ex) {
            ex.printStackTrace();
        }
        break;
    default:
        if (size < 2 || size > 3)
            System.out.println("unknown option");
    }
}

public static String headFile(File file, int lines) {
    RandomAccessFile fileHandler = null;
    try {
        fileHandler = new RandomAccessFile(file, "r" );
        StringBuilder stringBuilder = new StringBuilder();

        for(int i = 0; i < lines; i++) {
            stringBuilder.append(fileHandler.readLine()).append("\n");
        }

        String utf = new String(stringBuilder.toString()
            .getBytes(StandardCharsets.ISO_8859_1),
StandardCharsets.UTF_8);
        return utf;
    } catch ( java.io.FileNotFoundException e ) {
        e.printStackTrace();
        return null;
    } catch ( java.io.IOException e ) {
        e.printStackTrace();
        return null;
    }
    finally {
        if (fileHandler != null )
            try {
                fileHandler.close();
            } catch (IOException e) {
            }
        }
    }
}
}

```

Результат выполнения:

```
MacBook-Pro-Anastasia:lab2_2_jar anastasiakovaleva$ java -jar lab2-2.jar -n 5 file
You just finished an app. It works fine, but the interface lacks style and grace.
You could give it a makeover by drawing several sizes of custom control images
in Photoshop and hope Apple doesn't release a @4x retina screen. That plan, however,
lacks strategy and sounds time-consuming. Alternatively, you could use Core Graphics
to create an image that scales crisply for any device size.
```

```
MacBook-Pro-Anastasia:lab2_2_jar anastasiakovaleva$ java -jar lab2-2.jar 10 file
You just finished an app. It works fine, but the interface lacks style and grace.
You could give it a makeover by drawing several sizes of custom control images
in Photoshop and hope Apple doesn't release a @4x retina screen. That plan, however,
lacks strategy and sounds time-consuming. Alternatively, you could use Core Graphics
to create an image that scales crisply for any device size.
Core Graphics is Apple's vector drawing framework. It's a big, powerful application
programming interface (API) with many tools to master and cool features like
@IBDesignable and @IBInspectable. But never fear! This three-part series takes
a modern approach to Core Graphics. It starts slow and eases you in with fun,
engaging exercises. By the end, you'll be able to create stunning graphics for your apps.
```

Вывод: В ходе выполненной работы приобрела базовые навыки работы с файловой системой в Java.