

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ
“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”
КАФЕДРА ИНТЕЛЛЕКТУАЛЬНЫХ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЁТ
по лабораторной работе №2

Выполнила:
студентка группы ПО-9
Кот А. А.

Проверил:
Крощенко А. А.

Брест 2024

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

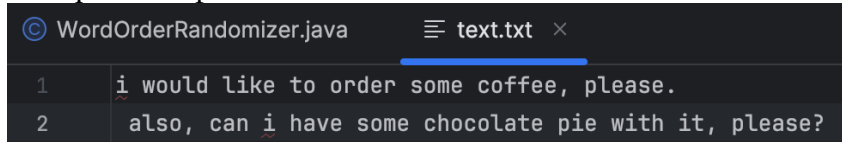
Вариант 8.

Ход работы

Задание 1.

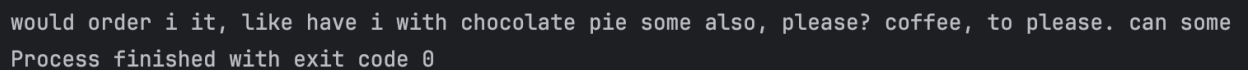
Напишите программу, считывающую текст построчно и изменяющую порядок следования слов на случайный. Строки с новым порядком слов выведите на экран.

Содержимое файла:



```
© WordOrderRandomizer.java  text.txt ×
1 i would like to order some coffee, please.
2 also, can i have some chocolate pie with it, please?
```

Вывод программы:



```
would order i it, like have i with chocolate pie some also, please? coffee, to please. can some
Process finished with exit code 0
```

Код программы:

WordOrderRandomizer.java

```
import java.io.*;
import java.util.Random;

public class WordOrderRandomizer {
    public static void main(String[] args) {
        randomiseWordOrder();
    }

    static void randomiseWordOrder(){
        String pathToFile = "src/text.txt";

        String[] wordsInText = readTextFromFile(pathToFile).split(" ");

        Random randomNum = new Random();
        for (int i = wordsInText.length - 1; i >= 0; i--) {
            int randomIndex = randomNum.nextInt(i + 1);
            String temp = wordsInText[randomIndex];
            wordsInText[randomIndex] = wordsInText[i];
            wordsInText[i] = temp;
        }

        StringBuilder randomisedText;
        randomisedText = new StringBuilder();
        for (String word : wordsInText) {
            randomisedText.append(word).append(" ");
        }

        System.out.print(randomisedText);
    }

    static String readTextFromFile (String pathToFile) {
        StringBuilder textFromFile = new StringBuilder();
        try (BufferedReader fileInput = new BufferedReader(new
        FileReader(pathToFile))) {
            String nextString;
            while ((nextString = fileInput.readLine()) != null) {
```

```

        textFromFile.append(nextString);
    }
} catch (FileNotFoundException e) {
    System.out.println("File not found!");
} catch (IOException e) {
    System.out.println("Reading was interrupted.");
}
return textFromFile.toString();
}
}

```

Задание 2.

Утилита paste выполняет слияние строк/столбцов из файлов и выводит результат в стандартный вывод.

Формат использования: `paste [options] [file1 [file2]..]`, где ключи имеют следующее значение:

- `-s` Меняет положение строк со столбцами;
- `-d` разделитель Меняет разделитель на указанный (по умолчанию TAB)

Работа программы:

`paste names.txt numbers.txt`

```

Mark Smith  555-1234
Bobby Brown 555-9876
Sue Miller  555-6743
Jenny Igo   876-5309

```

Process finished with exit code 0

`paste -s names.txt numbers.txt`

```

Mark Smith  Bobby Brown Sue Miller  Jenny Igo
555-1234    555-9876    555-6743    876-5309

```

Process finished with exit code 0

`paste -d ., names.txt numbers.txt`

```

Mark Smith.555-1234
Bobby Brown,555-9876
Sue Miller.555-6743
Jenny Igo,876-5309

```

Process finished with exit code 0

`paste -s -d '\t\n' names.txt`

```

Mark Smith  Bobby Brown
Sue Miller  Jenny Igo

```

Process finished with exit code 0

Код программы:

Paste.java

```
import java.io.*;
import java.util.*;

public class Paste {
    public static void main(String[] args) {
        int argsCount = args.length;
        if (argsCount < 1 || argsCount > 5) {
            System.out.println("Incorrect format. " +
                "Try: paste [-s][-d] [file1 [file2]..]");
            return;
        }

        boolean swap = false;
        List<Character> delimiter = new ArrayList<>();
        delimiter.add('\t');
        List<List<String>> textFromFiles = new ArrayList<>();

        for (int i = 0; i < argsCount; i++) {
            switch (args[i]) {
                case "-s":
                    swap = true;
                    break;
                case "-d":
                    if (!args[i + 1].contains(".txt")) {
                        delimiter = divideStringToDelimiters(args[i + 1]);
                        i++;
                    } else {
                        System.out.println("The delimiters " +
                            "were not provided!");
                        return;
                    }
                    break;
                default:
                    textFromFiles.add(readTextFromFile(args[i]));
            }
        }

        int i = 0;
        int delimiterCount = delimiter.size();

        if (textFromFiles.size() == 1) {
            for (String line : textFromFiles.get(0)) {
                if (swap) {
                    System.out.print(line +
                        delimiter.get(i % delimiterCount));
                } else {
                    System.out.println(line +
                        delimiter.get(i % delimiterCount));
                }
                i++;
            }
        } else {
            Iterator<String> iteratorFile1 = textFromFiles.get(0).iterator();
            Iterator<String> iteratorFile2 = textFromFiles.get(1).iterator();

            if (swap) {
                while (iteratorFile1.hasNext()) {
                    System.out.print(iteratorFile1.next() +
                        delimiter.get(i % delimiterCount));
                    i++;
                }
                System.out.println();
                while (iteratorFile2.hasNext()) {
```

```

        System.out.print(iteratorFile2.next() +
            delimiter.get(i % delimiterCount));
        i++;
    }
} else {
    while (iteratorFile1.hasNext() && iteratorFile2.hasNext()) {
        System.out.println(iteratorFile1.next()
            + delimiter.get(i % delimiterCount)
            + iteratorFile2.next());
        i++;
    }
}
}

static ArrayList<String> readTextFromFile (String pathToFile) {
    ArrayList<String> textFromFile = new ArrayList<>();
    try (BufferedReader fileInput = new BufferedReader(new
        FileReader(pathToFile))) {
        String nextString;
        while ((nextString = fileInput.readLine()) != null) {
            textFromFile.add(nextString);
        }
    } catch (FileNotFoundException e) {
        System.out.println("File not found!");
    } catch (IOException e) {
        System.out.println("Reading was interrupted.");
    }
    return textFromFile;
}

static List<Character> divideStringToDelimiters (String
    delimitersFromArg) {
    List<Character> arrayOfDelimiters = new ArrayList<>();
    for (int i = 0; i < delimitersFromArg.length(); i++) {
        char ch = delimitersFromArg.charAt(i);
        if (ch == '\\') {
            switch (delimitersFromArg.charAt(i + 1)) {
                case 'n':
                    arrayOfDelimiters.add('\n');
                    break;
                case 't':
                    arrayOfDelimiters.add('\t');
                    break;
                case 'r':
                    arrayOfDelimiters.add('\r');
                    break;
                default:
                    arrayOfDelimiters.add('\\');
                    i--;
                    break;
            }
            i++;
        } else {
            arrayOfDelimiters.add(ch);
        }
    }
    return arrayOfDelimiters;
}
}

```

Вывод: в ходе лабораторной работы были приобретены базовые навыки работы с файловой системой в Java.