

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:

Макаревич Е.С.

**Проверил:**

Крощенко А.А.

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

## Вариант 17

**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.

**Код программы:**

**task01.java**

```
package task01;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.HashMap;
import java.util.Map;
import java.util.Vector;

public class task01 {

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

        StringBuilder text = new StringBuilder();
        FileReader reader = new FileReader("src/task01/example.txt");
        BufferedReader br = new BufferedReader(reader);
        String line;

        int lineNumber = 1;
        while ((line = br.readLine()) != null) {
            text.append(line).append("\n");
            lineNumber++;
        }
        br.close();

        Vector<String> vec = new Vector<>();
        StringBuilder wordBuilder = new StringBuilder();
        lineNumber = 1;
        for (int i = 0; i < text.length(); i++) {
            char c = text.charAt(i);
            if (Character.isLetterOrDigit(c)) {
                wordBuilder.append(c);
            } else {
                if (wordBuilder.length() > 0) {
                    vec.add(wordBuilder.toString().toLowerCase() + ":" +
lineNumber);
                    wordBuilder.setLength(0);
                }
                if (c == '\n') {
                    lineNumber++;
                }
            }
        }
        if (wordBuilder.length() > 0) {
            vec.add(wordBuilder.toString().toLowerCase() + ":" + lineNumber);
        }
    }
}
```

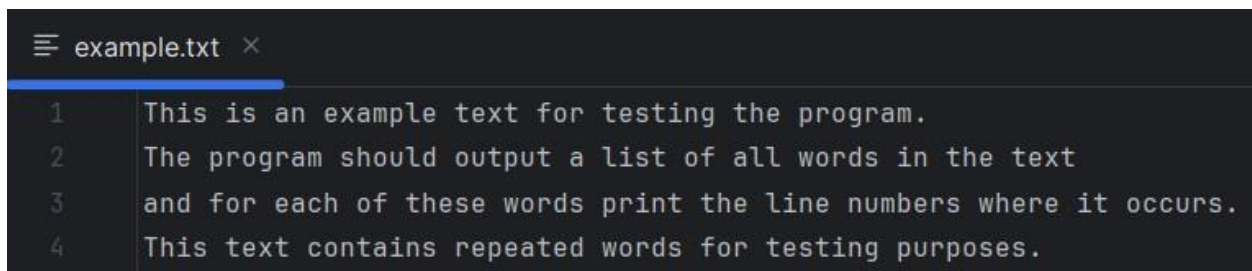
```

Map<String, Vector<Integer>> wordMap = new HashMap<>();
for (int i = 0; i < vec.size(); i++) {
    String[] parts = vec.get(i).split(":");
    String word = parts[0];
    int lineNum = Integer.parseInt(parts[1]);
    if (!wordMap.containsKey(word)) {
        wordMap.put(word, new Vector<>());
    }
    wordMap.get(word).add(lineNum);
}

for (Map.Entry<String, Vector<Integer>> entry : wordMap.entrySet()) {
    System.out.print(entry.getKey() + " contains in ");
    Vector<Integer> lineNumbers = entry.getValue();
    for (int i = 0; i < lineNumbers.size(); i++) {
        System.out.print(lineNumbers.get(i));
        if (i < lineNumbers.size() - 1) {
            System.out.print(", ");
        }
    }
    System.out.println();
}
}
}

```

### Входные данные:



```

1 This is an example text for testing the program.
2 The program should output a list of all words in the text
3 and for each of these words print the line numbers where it occurs.
4 This text contains repeated words for testing purposes.

```

### Результат программы:

```
line contains in 3
for contains in 1, 3, 4
numbers contains in 3
program contains in 1, 2
occurs contains in 3
repeated contains in 4
example contains in 1
output contains in 2
these contains in 3
and contains in 3
of contains in 2, 3
should contains in 2
where contains in 3
text contains in 1, 2, 4
all contains in 2
a contains in 2
in contains in 2
testing contains in 1, 4
this contains in 1, 4
words contains in 2, 3, 4
is contains in 1
it contains in 3
list contains in 2
an contains in 1
each contains in 3
the contains in 1, 2, 2, 3
print contains in 3
contains contains in 4
purposes contains in 4
```

**Задание 2.** Написать консольную утилиту, обрабатывающую ввод пользователя и дополнительные ключи. Проект упаковать в jar-файл, написать bat-файл для запуска.

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)

**Код программы:**

```
package task02;

import java.io.*;
```

```

import java.util.*;

public class task02 {
    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 index = 0;
        int delimiterCount = delimiter.size();

        if (textFromFiles.size() == 1) {
            for (String line : textFromFiles.get(0)) {
                if (swap) {
                    System.out.print(line +
                        delimiter.get(index % delimiterCount));
                } else {
                    System.out.println(line +
                        delimiter.get(index % delimiterCount));
                }
                index++;
            }
        } 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(index % delimiterCount));
                    index++;
                }
                System.out.println();
                while (iteratorFile2.hasNext()) {
                    System.out.print(iteratorFile2.next() +
                        delimiter.get(index % delimiterCount));

```

```

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

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;
}
}

```

**Входные данные:**

```

names.txt x
1 Mark Smith
2 Bobby Brown
3 Sue Miller
4 Jenny Igotit

```

```

numbers.txt x
1 555-1234
2 555-9876
3 555-6743
4 867-5309

```

## Результат программы:

Упаковали проект в jar-файл:

```

task02
├── lab2_2.jar
├── manifest.txt
├── names.txt
├── numbers.txt
├── paste.bat
├── task02
└── task02.class

```

```

D:\BrSTU\BrSTU\СПП\lab2\src\task02>paste names.txt numbers.txt
Mark Smith      555-1234
Bobby Brown     555-9876
Sue Miller      555-6743
Jenny Igotit    867-5309

```

```

D:\BrSTU\BrSTU\СПП\lab2\src\task02>paste -s names.txt numbers.txt
Mark Smith      Bobby Brown      Sue Miller      Jenny Igotit
555-1234        555-9876        555-6743       867-5309

```

```

D:\BrSTU\BrSTU\СПП\lab2\src\task02>paste -d ., names.txt numbers.txt
Mark Smith.555-1234
Bobby Brown,555-9876
Sue Miller.555-6743
Jenny Igotit,867-5309

```

```
D:\BrSTU\BrSTU\СПП\lab2\src\task02>paste -s -d \t\n names.txt
Mark Smith      Bobby Brown
Sue Miller      Jenny Igotit
```

**Вывод:** приобрели базовые навыки работы с файловой системой в Java.