



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных**

О Т Ч Е Т

по лабораторной работе № 7

Вариант 12

Название: Работа со строками и регулярными выражениями

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

П.А. Мартынюк

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2022

Цель работы:

Получение навыков работы со строками и регулярными выражениями в Java.

Выполнение:

Задание 1:

1. В русском тексте каждую букву заменить ее порядковым номером в алфавите. При выводе в одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
2. В тексте после буквы Р, если она не последняя в слове, ошибочно напечатана буква А вместо О. Внести исправления в текст.

Исходный текстовый файл для выполнения подзадачи 1 (файл text.txt)

```
Ночь, улица, фонарь, аптека,  
Бессмысленный и тусклый свет.  
Живи еще хоть четверть века -  
Все будет так. Исхода нет.  
Умрешь - начнешь опять сначала  
И повторится все, как встарь:  
Ночь, ледяная рябь канала,  
Аптека, улица, фонарь.
```

Листинг выполнения подзадачи 1 (файл LettersToNumbers.java)

```
package task1;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.io.IOException;  
import java.nio.charset.StandardCharsets;  
import java.nio.file.Files;  
import java.nio.file.Path;  
import java.nio.file.Paths;  
import java.nio.file.StandardOpenOption;  
import java.util.Collections;  
import java.util.Scanner;  
  
public class LettersToNumbers {  
    public static void main(String[] args) {  
  
        File inp_file = new  
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\text.txt");  
        Path out_file_path =  
Paths.get("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\text_with_let  
ter_numbers.txt");  
        File out_file = new  
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\text_with_letter_n  
umbers.txt");  
  
        if(out_file.delete()) {  
            try {  
                out_file.createNewFile();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
        }  
    }  
}
```

```

    }
}

Scanner scanner = null;
try {
    scanner = new Scanner(inp_file);
} catch (FileNotFoundException e) {
    e.printStackTrace();
}

String line_with_letters;
String line_with_numbers;

while(scanner.hasNextLine()) {
    String line = scanner.nextLine();
    char[] letters = line.toCharArray();

    line_with_letters = "";
    line_with_numbers = "";
    String space = "";

    for (int i = 0; i < letters.length; i++) {
        if(letters[i] >= 'a' && letters[i] <= 'я') {
            line_with_letters += (letters[i] + " ");
            space = ((letters[i] - 'a' + 1) < 10) ? " " : " ";
            line_with_numbers += (letters[i] - 'a' + 1) + space;
        } else if(letters[i] >= 'А' && letters[i] <= 'Я') {
            line_with_letters += (letters[i] + " ");
            space = ((letters[i] - 'А' + 1) < 10) ? " " : " ";
            line_with_numbers += (letters[i] - 'А' + 1) + space;
        } else {
            line_with_letters += letters[i] + " ";
            line_with_numbers += " ";
        }
    }

    try {
        Files.write(out_file_path, Collections.singleton(line_with_letters),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
        Files.write(out_file_path, Collections.singleton(line_with_numbers),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

}
}

```

Результирующий текстовый файл после выполнения подзадачи 1
(файл text_with_letter_numbers.txt)

```

Н о ч ь , у л и ц а , ф о н а р ь , а п т е к а ,
14 15 24 29      20 12 9  23 1      21 15 14 1  17 29      1 16 19 6  11 1
Б е с с м ы с л е н н ы й и т у с к л ы й с в е т .
2 6 18 18 13 28 18 12 6 14 14 28 10 9 19 20 18 11 12 28 10 18 3 6 19
Ж и в и е щ е х о т ь ч е т в е р т ь в е к а -
7 9 3 9 6 26 6 22 15 19 29 24 6 19 3 6 17 19 29 3 6 11 1
В с е б у д е т т а к . И с х о д а н е т .
3 18 6 2 20 5 6 19 19 1 11 9 18 22 15 5 1 14 6 19
У м р е ш ь - н а ч н е ш ь о п я т ь с н а ч а л а
20 13 17 6 25 29 14 1 24 14 6 25 29 15 16 32 19 29 18 14 1 24 1 12 1
И п о в т о р и т с я в с е , к а к в с т а р ь :
9 16 15 3 19 15 17 9 19 18 32 3 18 6 11 1 11 3 18 19 1 17 29
Н о ч ь , л е д я н а я р я б ь к а н а л а ,
14 15 24 29 12 6 5 32 14 1 32 17 32 2 29 11 1 14 1 12 1
А п т е к а , у л и ц а , ф о н а р ь .
1 16 19 6 11 1 20 12 9 23 1 21 15 14 1 17 29

```

Исходный текстовый файл для выполнения подзадачи 2 (файл text_with_mistakes.txt)

РАМАШКИ корава Радина

Листинг выполнения подзадачи 2 (файл LetterCorrection.java)

```
package task1;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.nio.file.StandardOpenOption;
import java.util.Collections;
import java.util.Scanner;

public class LetterCorrection {

    public static void main(String[] args) {

        File inp_file = new
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\text_with_mistakes
.txt");
        Path out_file_path =
Paths.get("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\corrected_tex
t.txt");
        File out_file = new
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task1\\corrected_text.txt
");

        if(out_file.delete()) {
            try {
                out_file.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (
            FileNotFoundException e) {
            e.printStackTrace();
        }

        while(scanner.hasNextLine()) {
            String line = scanner.nextLine();
            line = line.replaceAll("PA", "PO");
            line = line.replaceAll("pa", "po");
            line = line.replaceAll("Pa", "Po");
            line = line.replaceAll("pA", "pO");

            try {
                Files.write(out_file_path, Collections.singleton(line),
StandardCharsets.UTF_8, StandardOpenOption.APPEND);
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
}
```

Результирующий текстовый файл после выполнения подзадачи 2
(файл corrected_text.txt)

РОМАШКИ корова Родина

Задание 2:

1. Найти и напечатать, сколько раз повторяется в тексте каждое слово, которое встречается в нем.
2. В тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

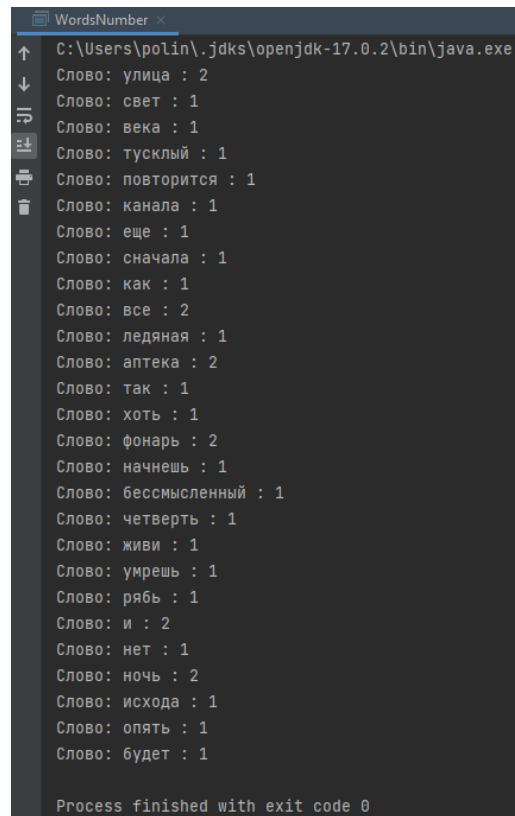
Исходный текстовый файл для выполнения подзадач 1 и 2 (файл text.txt)

```
Ночь, улица, фонарь, аптека,  
Бессмысленный и тусклый свет.  
Живи еще хоть четверть века -  
Все будет так. Исхода нет.  
Умрешь - начнешь опять сначала  
И повторится все, как встарь:  
Ночь, ледяная рябь канала,  
Аптека, улица, фонарь.
```

Листинг выполнения подзадачи 1 (файл WordsNumber.java)

```
package task2;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.HashMap;  
import java.util.Locale;  
import java.util.Scanner;  
import java.util.regex.Pattern;  
  
public class WordsNumber {  
    public static void main(String[] args) {  
  
        File inp_file = new  
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task2\\text.txt");  
  
        Scanner scanner = null;  
        try {  
            scanner = new Scanner(inp_file);  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        }  
  
        HashMap<String, Integer> dictionary = new HashMap<>();  
  
        while(scanner.hasNextLine()) {  
            String line = scanner.nextLine();  
            line = line.replace(" ", " ");  
            line = line.replace("\n", "");  
            line = line.toLowerCase(Locale.ROOT);  
            String[] words = line.split("[.?! , ]");  
  
            for (String word: words) {  
                if(Pattern.matches("[a-zA-Я]+", word)) {  
                    if (dictionary.containsKey(word)) {  
                        dictionary.put(word, dictionary.get(word) + 1);  
                    } else {  
                        dictionary.put(word, 1);  
                    }  
                }  
            }  
        }  
  
        for (String word : dictionary.keySet()) {  
            System.out.println("Слово: " + word + " : " + dictionary.get(word));  
        }  
    }  
}
```

```
}  
}
```



```
C:\Users\polin\.jdk\openjdk-17.0.2\bin\java.exe  
Слово: улица : 2  
Слово: свет : 1  
Слово: века : 1  
Слово: тусклый : 1  
Слово: повторится : 1  
Слово: канала : 1  
Слово: еще : 1  
Слово: сначала : 1  
Слово: как : 1  
Слово: все : 2  
Слово: ледяная : 1  
Слово: аптека : 2  
Слово: так : 1  
Слово: хоть : 1  
Слово: фонарь : 2  
Слово: начнешь : 1  
Слово: бессмысленный : 1  
Слово: четверть : 1  
Слово: живи : 1  
Слово: умрешь : 1  
Слово: рябь : 1  
Слово: и : 2  
Слово: нет : 1  
Слово: ночь : 2  
Слово: исхода : 1  
Слово: опять : 1  
Слово: будет : 1  
  
Process finished with exit code 0
```

Рисунок 1 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл CharNumber.java)

```
package task2;  
  
import java.io.File;  
import java.io.FileNotFoundException;  
import java.util.*;  
import java.util.regex.Pattern;  
  
public class CharNumber {  
    public static void main(String[] args) {  
        File inp_file = new  
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task2\\text.txt");  
  
        Scanner scanner = null;  
        try {  
            scanner = new Scanner(inp_file);  
        } catch (FileNotFoundException e) {  
            e.printStackTrace();  
        }  
  
        HashMap<Character, Integer> dictionary = new HashMap<>();  
  
        while(scanner.hasNextLine()) {  
            String line = scanner.nextLine();  
            line = line.replace(" ", " ");  
            line = line.replace("\n", "");  
            line = line.toLowerCase(Locale.ROOT);  
            char[] chars = line.toCharArray();  
  
            for (char ch: chars) {  
                if('a' <= ch && 'я' >= ch) {  
                    if (dictionary.containsKey(ch)) {  
                        dictionary.put(ch, dictionary.get(ch) + 1);  
                    } else {
```

```

        dictionary.put(ch, 1);
    }
}

Comparator<Map.Entry<Character, Integer>> valueComparator = new
Comparator<Map.Entry<Character, Integer>>() {
    @Override
    public int compare(Map.Entry<Character, Integer> e1, Map.Entry<Character, Integer>
e2) {
        int v1 = e1.getValue();
        int v2 = e2.getValue();
        return (v2 - v1);
    }
};

Set<Map.Entry<Character, Integer>> entries = dictionary.entrySet();
List<Map.Entry<Character, Integer>> listOfEntries = new ArrayList<Map.Entry<Character,
Integer>>(entries);
Collections.sort(listOfEntries, valueComparator);

System.out.println("Input number of letters: ");
Scanner in = new Scanner(System.in);
int n = in.nextInt();

for (int i = 0; i < n; i++) {
    System.out.println(listOfEntries.get(i));
}
}
}

```

Рисунок 2 - Результат выполнения кода решения подзадачи 2

Задание 3:

1. Найти наибольшее количество предложений текста, в которых есть одинаковые слова.
2. Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.

Текстовый файл для выполнения подзадач 1 и 2 (файл text.txt)

«Мастер и Маргарита» — роман Михаила Афанасьевича Булгакова, работа над которым началась в декабре 1928 года и продолжалась вплоть до смерти писателя. Роман относится к незавершённым произведениям; редактирование и сведение воедино черновых записей осуществляла после смерти мужа вдова писателя — Елена Сергеевна. Первая версия романа, имевшая названия «Копыто инженера», «Чёрный маг» и другие, была уничтожена Булгаковым в 1930 году. В последующих редакциях среди героев произведения появились автор романа о Понтии Пилате и его возлюбленная. Окончательное название — «Мастер и Маргарита» — оформилось в 1937 году.

Листинг выполнения подзадачи 1 (файл MaxSameWordsNumber.java)

```
package task3;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

public class MaxSameWordsNumber {
    public static void main(String[] args) {

        File inp_file = new
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task3\\text.txt");

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

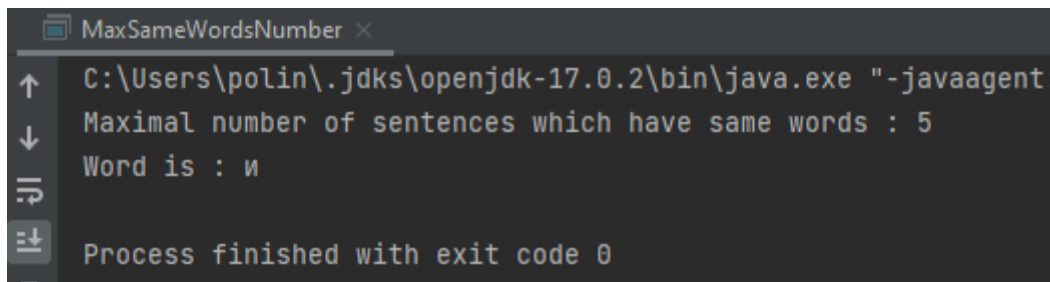
        HashMap<String, Integer> dictionary = new HashMap<>();
        String text = scanner.nextLine();
        String[] sentences = text.split("[.?!]");

        for (String sentence : sentences) {
            String[] words = sentence.split(" ");
            HashSet<String> wordsSet = new HashSet<>(Arrays.asList(words));
            for (String word : wordsSet) {
                if (dictionary.containsKey(word)) {
                    dictionary.put(word, dictionary.get(word) + 1);
                } else {
                    dictionary.put(word, 1);
                }
            }
        }

        Set<String> keys = dictionary.keySet();
        String maxPopularWord = sentences[0].split(" ")[0];

        for (String key : keys) {
            if (dictionary.get(maxPopularWord) < dictionary.get(key)) {
                maxPopularWord = key;
            }
        }

        System.out.print("Maximal number of sentences which have same words : ");
        System.out.println(dictionary.get(maxPopularWord));
        System.out.print("Word is : ");
        System.out.println(maxPopularWord);
    }
}
```

```
MaxSameWordsNumber X
C:\Users\polin\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:
Maximal number of sentences which have same words : 5
Word is : и
Process finished with exit code 0
```

Рисунок 3 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл FirstUniqueWord.java)

```
package task3;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

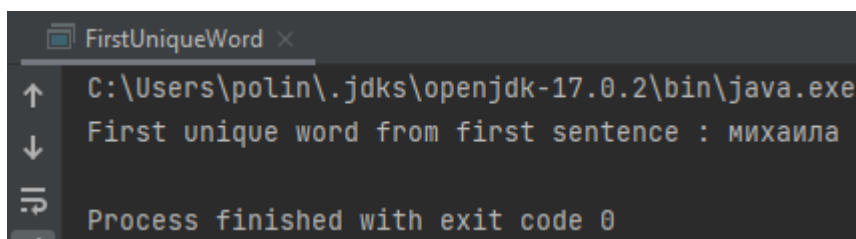
public class FirstUniqueWord {
    public static void main(String[] args) {
        File inp_file = new
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task3\\text.txt");

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        String text = scanner.nextLine();
        text = text.toLowerCase(Locale.ROOT);
        String[] sentences = text.split("[.?!]");
        String[] words = sentences[0].split("[<<-:]");

        boolean foundFlag;

        for (String word : words) {
            foundFlag = true;
            for (int i = 1; i < sentences.length; i++) {
                if(sentences[i].contains(word)) {
                    foundFlag = false;
                }
            }
            if(foundFlag){
                System.out.print("First unique word from first sentence : ");
                System.out.println(word);
                break;
            }
        }
    }
}
```



```
FirstUniqueWord X
C:\Users\polin\.jdk\openjdk-17.0.2\bin\java.exe
First unique word from first sentence : михаила
Process finished with exit code 0
```

Рисунок 4 - Результат выполнения кода решения подзадачи 2

Задание 4:

1. Ввести текст и список слов. Для каждого слова из заданного списка найти, сколько раз оно встречается в тексте, и рассортировать слова по убыванию количества вхождений.
2. Все слова текста рассортировать в порядке убывания их длин, при этом все слова одинаковой длины рассортировать в порядке возрастания в них количества гласных букв.

Текстовый файл для выполнения подзадач 1 и 2 (файл text.txt)

«Мастер и Маргарита» — роман Михаила Афанасьевича Булгакова, работа над которым началась в декабре 1928 года и продолжалась вплоть до смерти писателя. Роман относится к незавершённым произведениям; редактирование и сведение воедино черновых записей осуществляла после смерти мужа вдова писателя — Елена Сергеевна. Первая версия романа, имевшая названия «Копыто инженера», «Чёрный маг» и другие, была уничтожена Булгаковым в 1930 году. В последующих редакциях среди героев произведения появились автор романа о Понтии Пилате и его возлюбленная. Окончательное название — «Мастер и Маргарита» — оформилось в 1937 году.

Листинг выполнения подзадачи 1 (файл WordsNumberRating.java)

```
package task4;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.InputStream;
import java.util.*;
import java.util.regex.Pattern;

public class WordsNumberRating {
    public static void main(String[] args) {

        HashMap<String, Integer> dictionary = new HashMap<>();
        Scanner scanner = new Scanner(System.in);

        System.out.println("Input text:");
        String text = scanner.nextLine();
        text = text.toLowerCase(Locale.ROOT);
        String[] words = text.split("[.?!<»-:;]");

        for (String word: words) {
            if(Pattern.matches("[a-яА-Я]+", word)) {
                if (dictionary.containsKey(word)) {
                    dictionary.put(word, dictionary.get(word) + 1);
                } else {
                    dictionary.put(word, 1);
                }
            }
        }

        System.out.println("Input words:");
        String wordSetLine = scanner.nextLine();
        wordSetLine = wordSetLine.toLowerCase(Locale.ROOT);
        String[] wordAr = wordSetLine.split("[.?!<»-:;]");

        HashMap<String, Integer> wordSet = new HashMap<>();

        for (String word : wordAr) {
            if(dictionary.containsKey(word)){
                wordSet.put(word, dictionary.get(word));
            } else {
                wordSet.put(word, 0);
            }
        }
    }
}
```

```

    }
}

Comparator<Map.Entry<String, Integer>> valueComparator = new
Comparator<Map.Entry<String, Integer>>() {
    @Override
    public int compare(Map.Entry<String, Integer> e1, Map.Entry<String, Integer> e2) {
        int v1 = e1.getValue();
        int v2 = e2.getValue();
        return (v2 - v1);
    }
};

Set<Map.Entry<String, Integer>> entries = wordSet.entrySet();
List<Map.Entry<String, Integer>> listOfEntries = new ArrayList<Map.Entry<String,
Integer>>(entries);
Collections.sort(listOfEntries, valueComparator);

System.out.println(listOfEntries);
}
}

```

```

C:\Users\polin\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021
Input text:
«Мастер и Маргарита» — роман Михаила Афанасьевича Булгакова, работа над которым началась в декабре 1928 г
Input words:
мастер роман работа
[мастер=2, роман=2, работа=1]
Process finished with exit code 0

```

Рисунок 5 - Результат выполнения кода решения подзадачи 1

Листинг выполнения подзадачи 2 (файл WordsSort.java)

```

package task4;

import java.io.File;
import java.io.FileNotFoundException;
import java.util.*;

public class WordsSort {
    public static void main(String[] args) {

        File inp_file = new
File("C:\\Users\\polin\\IdeaProjects\\JavaMagisterCourse\\Lab7\\src\\task4\\text.txt");

        Scanner scanner = null;
        try {
            scanner = new Scanner(inp_file);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }

        String glasniye = "УуЕеЫыАаОоЭэЁёЯяИиЮю";

        String text = scanner.nextLine();
        text = text.toLowerCase(Locale.ROOT);
        text = text.replace(" ", " ");
        ArrayList<String> words = new ArrayList<>(Arrays.asList(text.split("[.?!,<<>-:;]")));

        Comparator<String> valueComparator = new Comparator<String>() {
            @Override
            public int compare(String e1, String e2) {
                if (e1.length() > e2.length()) {
                    return -1;
                } else if (e1.length() < e2.length()) {
                    return 1;
                } else {
                    int count1 = 0;

```

```

        int count2 = 0;
        for (int i = 0; i < e1.length(); i++) {
            if (glasniye.contains(e1.substring(i, i + 1))) {
                count1++;
            }
        }
        for (int i = 0; i < e2.length(); i++) {
            if (glasniye.contains(e2.substring(i, i + 1))) {
                count2++;
            }
        }
        return (count1 - count2);
    }
}

};

words.removeAll(Collections.singleton(""));
Collections.sort(words, valueComparator);
System.out.println(words);
}
}

```

```

WordsSort x
C:\Users\polin\.jdk\openjdk-17.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\lib\idea_rt.jar=64610:C:\Program Files\JetBrains\IntelliJ IDEA 2021.3.2\bin" -Dfile.encoding=UTF-8 -classpath C:\Users\polin\IdeaProjects\JavaMagisterCourse\out\production\Lab7\task4.WordsSort
[редактирование, незавершённым, произведениям, окончательное, продолжалась, осуществляла, возлюбленная, афанасьевича, произведения, последующих, булгаковым, оформилось, уничтожена, маргарита, булгакова, относится, сергеевна, редакциях, появились, маргарита, началась, черновых, писателя, сведение, писателя, названия, инженера, название, которым, декабре, записей, михаила, воедино, имевшая, вплоть, мастер, смерти, смерти, чёрный, мастер, работа, первая, версия, романа, копыто, другие, героев, романа, понтии, пилате, роман, роман, после, вдова, среди, автор, елена, 1928, 1930, 1937, года, мужа, была, году, году, над, маг, его, до, в, к, в, в, в, и, и, и, и, о, и, и]

```

Рисунок 6 - Результат выполнения кода решения подзадачи 2

Ссылка на программное решение:

Программное решение представлено в репозитории распределённой системы управления версиями Git:

<https://github.com/Owlfeather/JavaMagisterCourse/tree/main/Lab7/src>

Вывод:

При выполнении лабораторной работы были получены навыки работы со строками и регулярными выражениями в Java.