



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 09.04.01/12 Интеллектуальный анализ больших
данных в системах поддержки принятия решений.

О Т Ч Е Т

по лабораторной работе № 7

Вариант № 11

Название: Строки и регулярные выражения

Дисциплина: Языки программирования для работы с большими данными

Студент

ИУ6-23М

(Группа)

(Подпись, дата)

С.В.Мельников

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

(И.О. Фамилия)

Москва, 2023

Цель работы

Изучить работу со строками и применение регулярных выражений в языке программирования Java.

Задание 1 (Вариант 1, Задание 1):

В каждом слове текста k-ю букву заменить заданным символом. Если k больше длины слова, корректировку не выполнять.

Листинг программы:

Код класса ReplaceKLetter:

```
public class ReplaceKLetter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String[] text = scanner.nextLine().trim().split("\\s+");
        // String[] text = ("My name is Dima. I am a student of the second year of
        study. I study at the university. " +
        // "My hobby is music. When I have free time I usually go to the
        biggest music shop in my town and " +
        // "buy CDs with my favourite music. I have a big collection of CDs
        at home, but I can not resist " +
        // "the temptation of buying a new CD. That what I like to buy most
        of all.").trim().split("\\s+");
        System.out.print("Введите номер буквы в слове: ");
        int n = scanner.nextInt();
        System.out.print("Введите символ: ");
        char ch = scanner.next().charAt(0);
        for (int i = 0; i < text.length; i++) {
            if (text[i].length() > n) {
                char[] chars = text[i].toCharArray();
                for (int j = 0; j < chars.length; j++) {
                    if ((j + 1) % n == 0) {
                        chars[j] = ch;
                    }
                }
                text[i] = new String(chars);
            }
        }
        System.out.println("Текст после замены букв:");
        for (String s : text) {
            System.out.print(s + " ");
        }
    }
}
```

Работа программы представлена на рисунке 1.

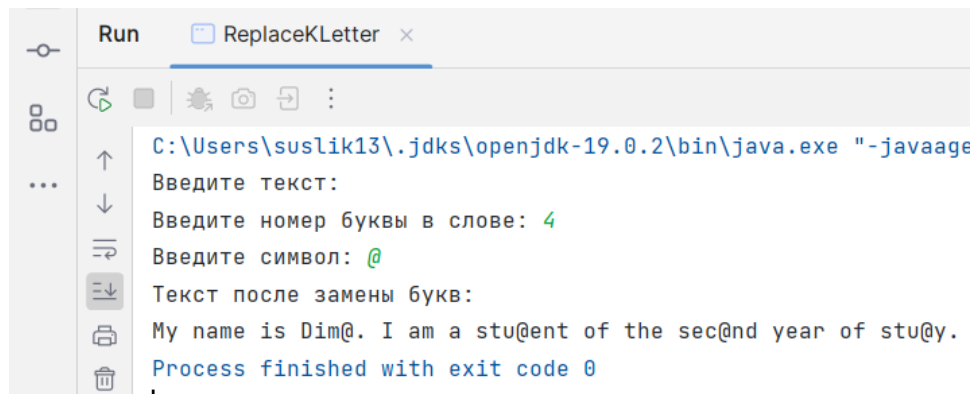


Рисунок 1 – Работа программы ReplaceKLetter.java

Задание 2 (Вариант 1, Задание 2):

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.

Листинг программы:

Код класса RusAlphabet:

```
public class RusAlphabet {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String text = scanner.nextLine();
        // String text = "Привет мир! Это строка на русском.";
        StringBuilder text_array = new StringBuilder();
        StringBuilder nums_array = new StringBuilder();
        for (char ch: text.toCharArray()) {
            text_array.append(" ").append(ch);
            int num = 0;
            if ('a' <= (int)ch && (int)ch <= 'я' || (int)ch == 'ё') {
                if ((int) ch == (int) 'ё') {
                    num = 7;
                } else if ((int) ch >= 'ж') {
                    num = (int) ch - 'a' + 2;
                } else {
                    num = (int) ch - 'a' + 1;
                }
            } else if ('A' <= (int)ch && (int)ch <= 'Я' || (int)ch == 'Ё') {
                if ((int) ch == (int) 'Ё') {
                    num = 7;
                } else if ((int) ch >= 'Ж') {
                    num = (int) ch - 'A' + 2;
                } else {
                    num = (int) ch - 'A' + 1;
                }
            }
            String str_num;
            if (num == 0) {
                str_num = " ";
            }
        }
    }
}
```

```

    } else {
        str_num = String.format("%3d", num);
    }
    nums_array.append(str_num);
}
System.out.println("Текст, разбитый по буквам и номера букв:");
System.out.println(text_array);
System.out.println(nums_array);
}
}

```

Работа программы представлена на рисунке 2.



Рисунок 2 – Работа программы RusAlphabet.java

Задание 3 (Вариант 2, Задание 1):

В тексте найти все пары слов, из которых одно является обращением другого.

Листинг программы:

Код класса Palindromes:

```

public class Palindromes {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String initial_text = scanner.nextLine();
        // String initial_text = "Идем молод долом меди меди тот";
        String[] text = initial_text.replaceAll("[./,\"()]:]",
        "").toLowerCase().trim().split("\\s+");
        int[] is_used = new int[text.length];
        List<Pair<String, String>> pairs = new ArrayList<>();
        for (int i = 0; i < text.length; i++) {
            for (int j = 0; j < text.length; j++) {
                StringBuilder word = new StringBuilder(text[j]);
                if (text[i].equals(word.reverse().toString()) && is_used[i] == 0
&& is_used[j] == 0 && i != j) {
                    pairs.add(new Pair<>(text[i], text[j]));
                    is_used[i] = 1;
                    is_used[j] = 1;
                }
            }
        }
    }
}

```

```

        System.out.println("Все пары слов:");
        for (Pair<String, String> pair: pairs) {
            System.out.println(pair.first + " " + pair.second);
        }
    }
}

```

Работа программы представлена на рисунке 3.

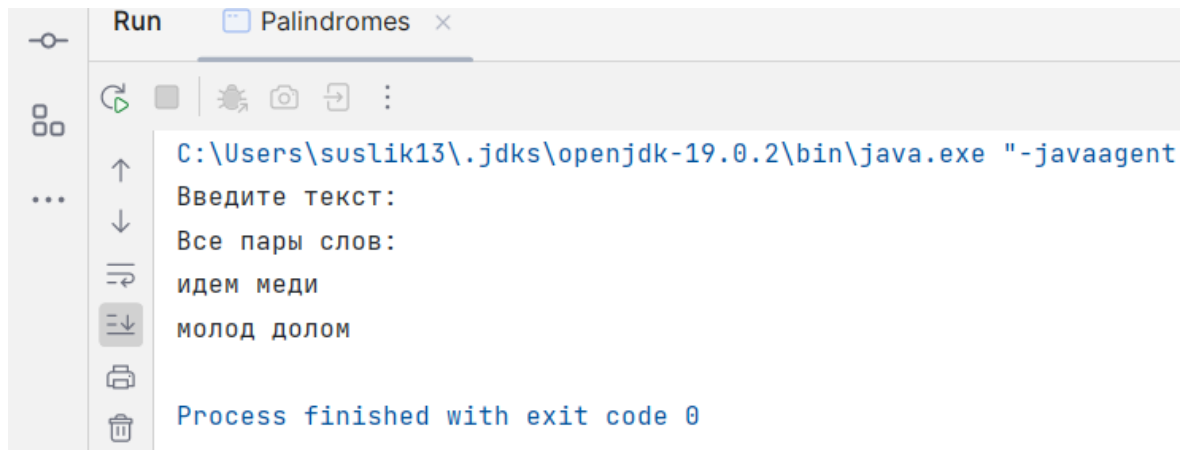


Рисунок 3 – Работа программы Palindromes.java

Задание 4 (Вариант 2, Задание 2):

Найти и напечатать, сколько раз повторяется в тексте каждое слово, которое встречается в нем.

Листинг программы:

Код класса WordsCounter:

```

public class WordsCounter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String initial_text = scanner.nextLine();
        // String initial_text = "Идем молод долом меди меди тот";
        String[] text = initial_text.replaceAll("[/.,\\\"() :]",
            "").toLowerCase().trim().split("\\s+");
        Map<String, Integer> words = new HashMap<>();
        for (String word: text) {
            Integer count = words.get(word);
            if(count == null) {
                count = 0;
            }
            words.put(word, ++count);
        }
        System.out.println("Итого слов:");
        for(String word : words.keySet()) {

```

```

        System.out.println(word + ": " + words.get(word));
    }
}

```

Работа программы представлена на рисунке 4.

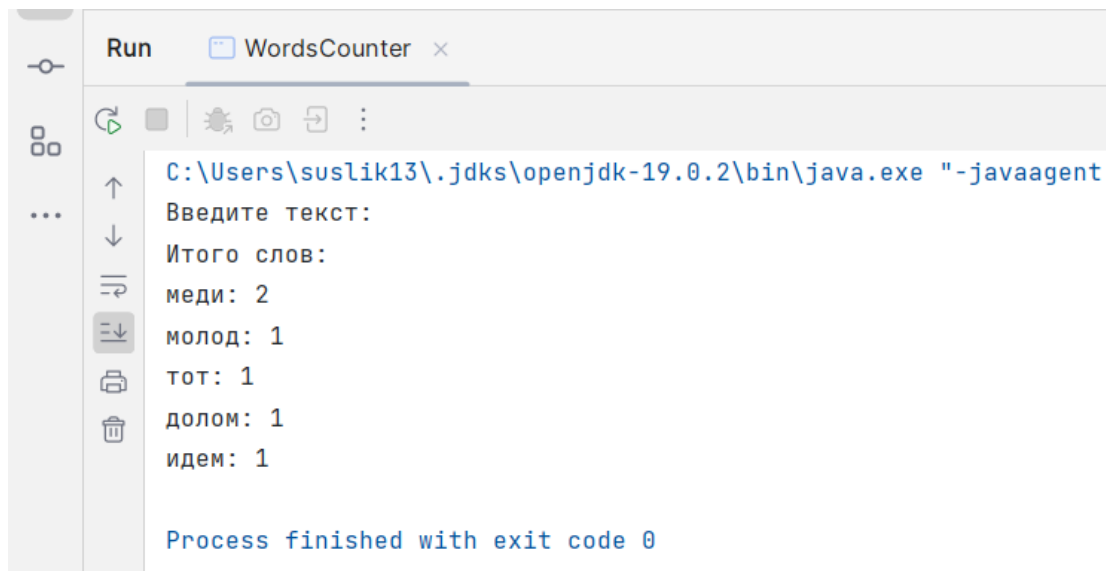


Рисунок 4 – Работа программы WordsCounter.java

Задание 5 (Вариант 3, Задание 1):

В тексте нет слов, начинающихся одинаковыми буквами. Напечатать слова текста в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой последующего слова. Если все слова нельзя напечатать в таком порядке, найти такую цепочку, состоящую из наибольшего количества слов.

Листинг программы:

Код класса WordsChain:

```

public class WordsChain {

    public static int next_word(String[] words, String word) {
        for (int i = 0; i < words.length; i++)
            if (words[i].charAt(0) == word.charAt(word.length() - 1))
                return i;
        return -1;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
    }
}

```

```

        String initial_text = scanner.nextLine();
//        String initial_text = "Hello abc world, Oliver down night clone Eliot
radius trello";
        String[] text = initial_text.replaceAll("[/.,\\\"() :]",
        "").toLowerCase().trim().split("\\s+");
        int start = 0;
        int max_len = 0;
        for (int i = 0; i < text.length; i++) {
            int count = 1;
            String word = text[i];
            int next_word_pos = next_word(text, word);
            while (next_word_pos != -1) {
                count++;
                word = text[next_word_pos];
                next_word_pos = next_word(text, word);
            }
            if (count > max_len) {
                max_len = count;
                start = i;
            }
        }
        int next = next_word(text, text[start]);
        System.out.print(text[start] + " ");
        while (next != -1) {
            System.out.print(text[next] + " ");
            next = next_word(text, text[next]);
        }
        System.out.println();
    }
}

```

Работа программы представлена на рисунке 5.

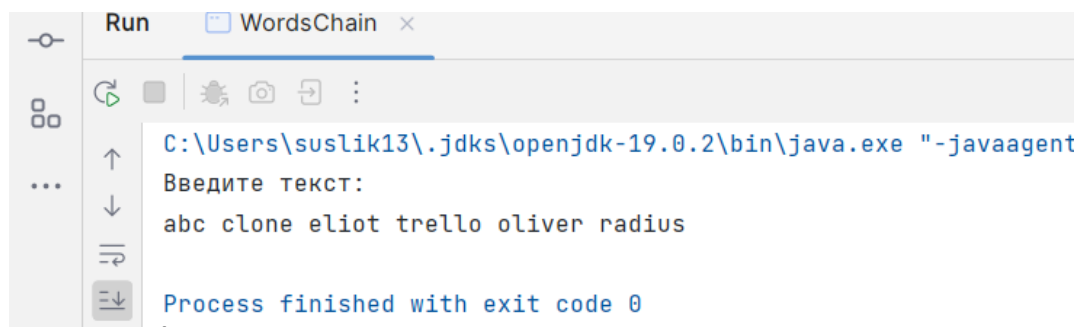


Рисунок 5 – Работа программы WordsChain.java

Задание 6 (Вариант 3, Задание 2):

Найти наибольшее количество предложений текста, в которых есть одинаковые слова.

Листинг программы:

Код класса FrequentWords:

```

public class FrequentWords {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String initial_text = scanner.nextLine();
        // String initial_text = ""
        // It goes without saying that humans (mammals identifiable as
        those that stand upright and are comparatively advanced and capable of detailed
        thought) have pretty remarkable bodies, given all that they've accomplished.
        Furthermore, an especially intelligent human brain produced this text! To be sure,
        humans have overcome predators, disease, and all sorts of other obstacles over
        thousands of years.
        // To fully understand and appreciate these accomplishments, let's
        take at some of the most well-known parts of the human body!
        // The head, or the spherical body part that contains the brain and
        rests at the top of the human body, has quite a few individual organs and body
        parts on it. It should quickly be mentioned that hair occupies the space on top of
        the head, and the ears, the organs responsible for hearing, are located on either
        side of the head. From top to bottom, the eyebrows, or horizontal strips of hair
        that can be found above the eye, are the first components of the head. The eyes
        are below them, and are round, orb-like organs that allow humans to see.
        // The eyes make way for the nose, or an external (sticking-out)
        organ that plays an important part in the breathing and bacteria-elimination
        processes. Below that is the mouth, or a wide, cavernous organ that chews food,
        removes bacteria, helps with breathing, and more. The mouth contains teeth, or
        small, white-colored, pointed body parts used to chew food, and the tongue, or a
        red-colored, boneless organ used to chew food and speak.
        // The neck is the long body part that connects the head to the
        chest (the muscular body part that protects the heart and lungs), and the stomach,
        or the part of the body that contains food and liquid-processing organs, comes
        below that.
        // The legs are the long, muscular body parts that allow humans to
        move from one spot to another and perform a variety of actions. Each leg contains
        a thigh (a thick, especially muscular body part used to perform strenuous motions;
        the upper part of the leg) and a calf (thinner, more flexible body part that
        absorbs the shock associated with movement; the lower part of the leg). Feet can
        be found at the bottom of legs, and each foot is comprised of five toes, or small
        appendages that help balance.
        // Arms are long, powerful body parts that are located on either
        side of chest, below the shoulders; arms are comprised of biceps (the thicker, more
        powerful upper portion), and forearms (the thinner, more flexible lower portion).
        Hands, or small, gripping body parts used for a tremendous number of actions, are
        at the end of arms. Each hand contains five fingers, or small appendages used to
        grip objects.
        // The aforementioned shoulders are rounded body parts that aid
        arms' flexibility. One's back is found on the opposite side of the stomach, and is
        a flat section of the body that contains important muscles that're intended to
        protect the lungs and other internal organs, in addition to helping humans perform
        certain motions and actions.""";
        String[] sentences = initial_text.split("[./!?]\\s*");
        Map<String, Set<Integer>> invert_index = new HashMap<>();
        for (int i = 0; i < sentences.length; i++) {
            String[] words = sentences[i].replaceAll("[,\\\"() :]", "").split("\\s");
            for (String word: words) {
                if (invert_index.get(word) == null) {
                    Set<Integer> set = new TreeSet<>();
                    invert_index.put(word, set);
                }
                invert_index.get(word).add(i);
            }
        }
        Set<Integer> max_set = new TreeSet<>();
        StringBuilder frequent_word = new StringBuilder();
        for (Map.Entry<String, Set<Integer>> curr: invert_index.entrySet()) {
            if (curr.getValue().size() > max_set.size()) {

```



```

        max_set = curr.getValue();
        frequent_word = new StringBuilder(curr.getKey());
    } else if (curr.getValue().size() == max_set.size() &&
curr.getValue().equals(max_set)) {
        frequent_word.append(", ").append(curr.getKey());
    }
}
System.out.println("Самые частые слова: " + frequent_word);
for (int index: max_set) {
    System.out.println(index + ": " + sentences[index]);
}
}
}

```

Работа программы представлена на рисунке 6.

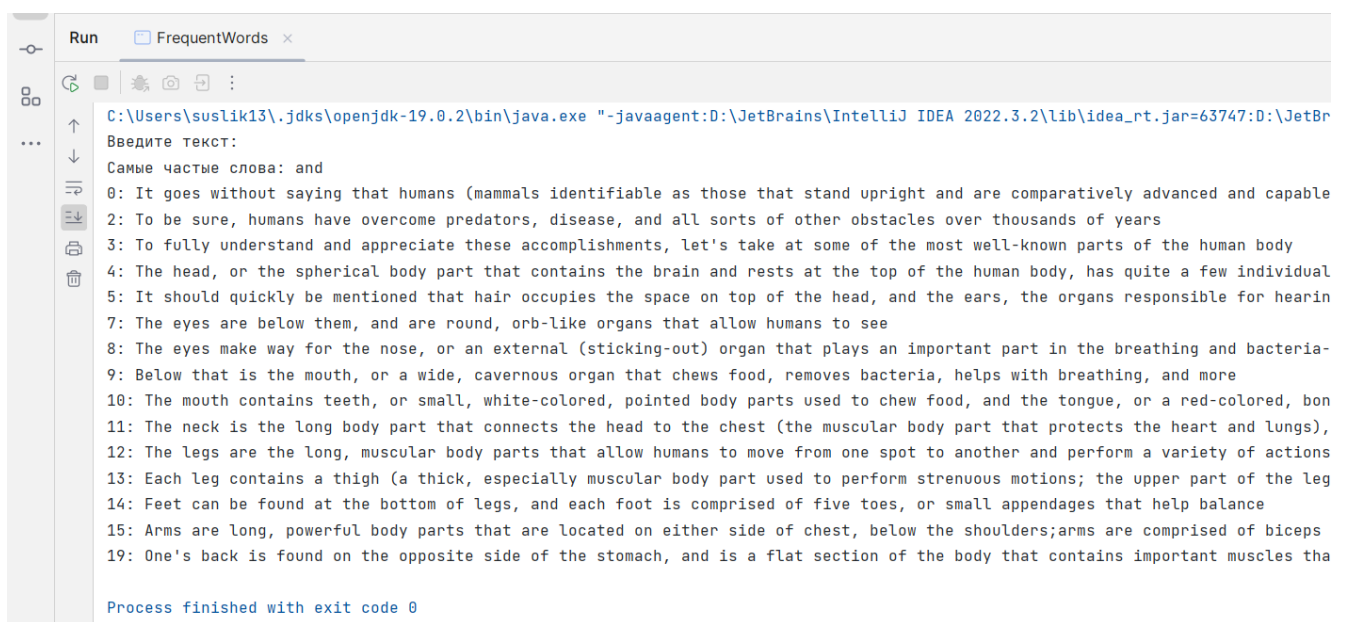


Рисунок 6 – Работа программы FrequentWords.java

Задание 7 (Вариант 4, Задание 1):

Все слова английского текста рассортировать по возрастанию количества заданной буквы в слове. Слова с одинаковым количеством расположить в алфавитном порядке.

Листинг программы:

Код класса SortByLetter:

```

public class SortByLetter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
    }
}

```

```

        String text = scanner.nextLine();
//        String text = "crab apple abacaba allocator all aaaaaa aerobus abroad
actually ago age";
        String[] words = text.split(" ");
        System.out.print("Введите букву: ");
        char ch = scanner.next().charAt(0);
        Arrays.sort(words, (s1, s2) -> {
            int count1 = 0;
            int count2 = 0;
            for (int i = 0; i < s1.length(); i++) {
                if (s1.charAt(i) == ch) {
                    count1 += 1;
                }
            }
            for (int i = 0; i < s2.length(); i++) {
                if (s2.charAt(i) == ch) {
                    count2 += 1;
                }
            }
            if (count1 > count2)
                return 1;
            else if (count1 == count2)
                return s1.compareTo(s2);
            else
                return -1;
        });
        for (String word: words) {
            System.out.print(word + " ");
        }
        System.out.println();
    }
}

```

Работа программы представлена на рисунке 7.



Рисунок 7 – Работа программы SortByLetter.java

Задание 8 (Вариант 4, Задание 2):

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.

Листинг программы:

Код класса FindWords:

```
public class FindWords {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст:");
        String initial_text = scanner.nextLine();
        // String initial_text = ""
        // It goes without saying that humans (mammals identifiable as
        those that stand upright and are comparatively advanced and capable of detailed
        thought) have pretty remarkable bodies, given all that they've accomplished.
        Furthermore, an especially intelligent human brain produced this text! To be sure,
        humans have overcome predators, disease, and all sorts of other obstacles over
        thousands of years.
        // To fully understand and appreciate these accomplishments, let's
        take at some of the most well-known parts of the human body!
        // The head, or the spherical body part that contains the brain and
        rests at the top of the human body, has quite a few individual organs and body
        parts on it. It should quickly be mentioned that hair occupies the space on top of
        the head, and the ears, the organs responsible for hearing, are located on either
        side of the head. From top to bottom, the eyebrows, or horizontal strips of hair
        that can be found above the eye, are the first components of the head. The eyes
        are below them, and are round, orb-like organs that allow humans to see.
        // The eyes make way for the nose, or an external (sticking-out)
        organ that plays an important part in the breathing and bacteria-elimination
        processes. Below that is the mouth, or a wide, cavernous organ that chews food,
        removes bacteria, helps with breathing, and more. The mouth contains teeth, or
        small, white-colored, pointed body parts used to chew food, and the tongue, or a
        red-colored, boneless organ used to chew food and speak.
        // The neck is the long body part that connects the head to the
        chest (the muscular body part that protects the heart and lungs), and the stomach,
        or the part of the body that contains food and liquid-processing organs, comes
        below that.
        // The legs are the long, muscular body parts that allow humans to
        move from one spot to another and perform a variety of actions. Each leg contains
        a thigh (a thick, especially muscular body part used to perform strenuous motions;
        the upper part of the leg) and a calf (thinner, more flexible body part that
        absorbs the shock associated with movement; the lower part of the leg). Feet can
        be found at the bottom of legs, and each foot is comprised of five toes, or small
        appendages that help balance.
        // Arms are long, powerful body parts that are located on either
        side of chest, below the shoulders; arms are comprised of biceps (the thicker, more
        powerful upper portion), and forearms (the thinner, more flexible lower portion).
        Hands, or small, gripping body parts used for a tremendous number of actions, are
        at the end of arms. Each hand contains five fingers, or small appendages used to
        grip objects.
        // The aforementioned shoulders are rounded body parts that aid
        arms' flexibility. One's back is found on the opposite side of the stomach, and is
        a flat section of the body that contains important muscles that're intended to
        protect the lungs and other internal organs, in addition to helping humans perform
        certain motions and actions.""
        String[] words = initial_text.replaceAll("[\\s()]",
        "").toLowerCase().split("\\s+");
```

```

System.out.println("Введите список слов:");
String[] words_list = scanner.nextLine().split(" ");
// String[] words_list = {"the", "is", "body", "are"};
Map<String, Integer> counter = new HashMap<>();
for (String word : words_list) {
    counter.put(word, 0);
}
for (String word : words) {
    if (counter.get(word) == null)
        continue;
    counter.replace(word, counter.get(word) + 1);
}
List<Pair<String, Integer>> sorted_counter = new ArrayList<>();
for (Map.Entry<String, Integer> word_count: counter.entrySet()) {
    sorted_counter.add(new Pair<>(word_count.getKey(),
word_count.getValue()));
}
sorted_counter.sort((p1, p2) -> {
    if (p2.second - p1.second == 0) {
        return p1.first.compareTo(p2.first);
    } else {
        return p2.second - p1.second;
    }
});
for (Pair<String, Integer> sorted_word: sorted_counter) {
    System.out.println(sorted_word.first + ": " + sorted_word.second);
}
}
}

```

Работа программы представлена на рисунке 8.

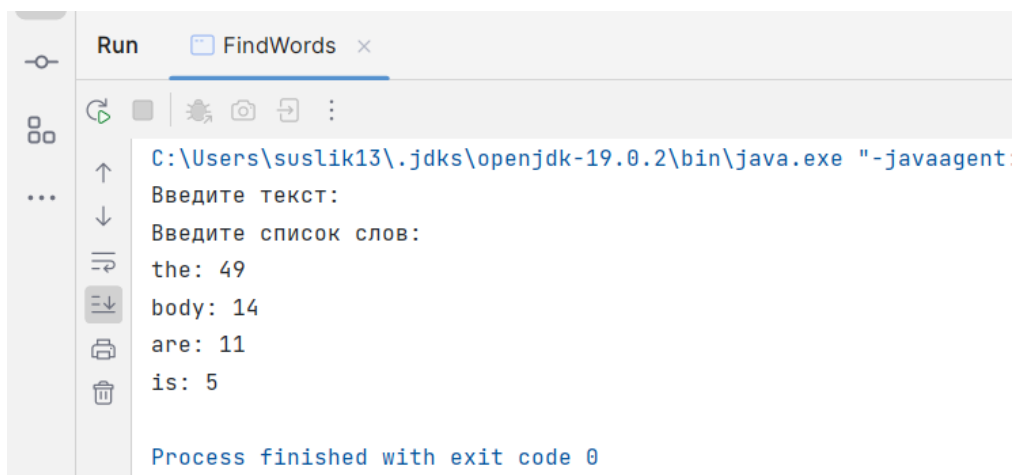


Рисунок 8 – Работа программы FindWords.java

Вывод

В ходе выполнения лабораторной работы были получены навыки работы со строками и регулярными выражениями в языке программирования Java.