



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
НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**
МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа,
обработки и интерпретации больших данных.**

Отчет

по лабораторной работе №7

Дисциплина: Языки программирования для работы с большими данными.

Студент

ИУ6-23М

(Группа)

В.А Антонов

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

П.В. Степанов

(Подпись, дата)

(И.О. Фамилия)

Москва, 2022

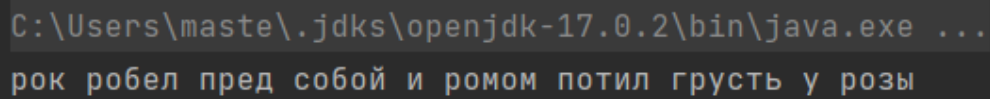
Лабораторная работа №7

Задание: Вариант 1

3. В тексте после буквы Р, если она не последняя в слове, ошибочно напечатана буква А вместо О. Внести исправления в текст.

Ход работы: Код программы файла Task_1_3

```
public class Task_1_3 {  
    public static void main(String[] args) {  
        String s = "рак рабел пред собой и рамом потил грусть у разы";  
        System.out.println(s.replaceAll("pa", "po"));  
    }  
}
```



```
C:\Users\maste\jdk\openjdk-17.0.2\bin\java.exe ...  
рок робел пред собой и ромом потил грусть у розы
```

Рисунок 1. Результат работы программы

4. В тексте слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.

Код программы файла Task_1_4

```
import java.util.Arrays;  
import java.util.Scanner;  
import java.util.stream.Collectors;  
  
public class Task_1_4 {  
    public static String replaceByLen(String sourceStr, int wordToReplaceLen,  
String newStr) {  
        return Arrays.stream(sourceStr.split(" ")) //Стрим из массива  
            .map(str -> str.length() == wordToReplaceLen ? newStr : str)  
            .collect(Collectors.joining(" "));  
    }  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(System.in);  
        System.out.println("STR: ");  
        String str = scanner.nextLine();  
        System.out.println("Введите n");  
        int n = scanner.nextInt();  
        System.out.println("Заменить на ");  
        String newWord;  
        scanner.nextLine();  
        newWord = scanner.nextLine();  
        System.out.println(replaceByLen(str, n, newWord));  
    }  
}
```

```

STR:
ва апапрап апрап ра апрапара
Введите n
2
Заменить на
вавпварваасмпирьвыаыва
вавпварваасмпирьвыаыва апапрап апрап вавпварваасмпирьвыаыва апрапара

```

Рисунок 2. Результат работы программы

Задание: Вариант 2

3. В тексте найти и напечатать n символов (и их количество), встречающихся наиболее часто.

Ход работы: Код программы файла Task_2_3

```

import java.util.*;

public class Task_2_3 {
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст");
        String str = scanner.nextLine();

        HashMap<Character, Integer> hm = new HashMap<>(); // символ-число
        for (int i = 0; i < str.length(); i++) {
            if (hm.get(str.charAt(i)) != null) { // получаем значение char
                int temp = hm.get(str.charAt(i));
                hm.put(str.charAt(i), ++temp); // С помощью метода put()
добавляем в хеш-отображение пару
            } else {
                hm.put(str.charAt(i), 1);
            }
        }

        Map<Character, Integer> result = new LinkedHashMap<>(); // хранить
данные в порядке добавления
        hm.entrySet().stream()
            //Возвращает компаратор, который сравнивает Map.Entry в
естественном порядке по значению.
            .sorted(Map.Entry.<Character,
Integer>comparingByValue().reversed())
            // результат
            .forEachOrdered(x -> result.put(x.getKey(), x.getValue()));

        System.out.println(result);
    }
}

```

Введите текст

ывывывыв ывывыва ывывпп ывврп

{в=8, а=6, ы=6, п=5, =3, р=1}

Рисунок 3. Результат работы программы

4. Найти, каких букв, гласных или согласных, больше в каждом предложении текста.

Код программы файла Task_2_4

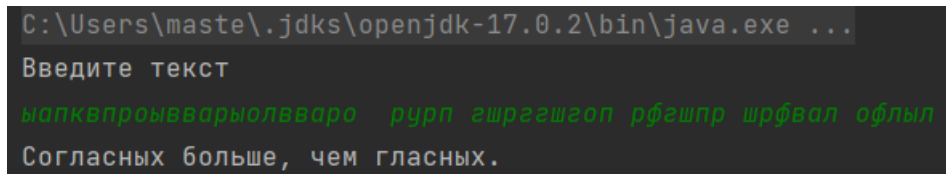
```
import java.util.Arrays;
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;

public class Task_2_4 {

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Введите текст");
        String str = scanner.nextLine();
        // HashSet это одно множество объектов
        HashSet<Character> vowels =
            new HashSet<>(
                Arrays.asList('а', 'о', 'э', 'е', 'и', 'ы', 'у', 'ё',
'ю', 'я'));// метод принимает массив в качестве параметра и возвращает
список
        HashSet<Character> consonants =
            new HashSet<>(
                Arrays.asList('б', 'в', 'г', 'д', 'ж', 'з', 'й', 'к',
'л', 'м', 'н', 'п', 'р', 'с', 'т', 'ф', 'х', 'ц', 'ч', 'ш', 'щ'));

        int numberOfVowels = 0;
        int numberOfConsonants = 0;
        for (int i = 0; i < str.length(); i++) {
            char ch = Character.toLowerCase(str.charAt(i)); // Преобразует
символ в нижний регистр
            if (vowels.contains(ch)) { //Проверка наличие объекта
                numberOfVowels++;
            } else if (consonants.contains(ch)) {
                numberOfConsonants++;
            }
        }

        if (numberOfVowels > numberOfConsonants) {
            System.out.println("Гласных больше, чем согласных.");
        } else {
            System.out.println("Согласных больше, чем гласных.");
        }
    }
}
```



```
C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...  
Введите текст  
напквпроивварыоляваро рурп гшрегепоп рфгшпр шрфвал офлвл  
Согласных больше, чем гласных.
```

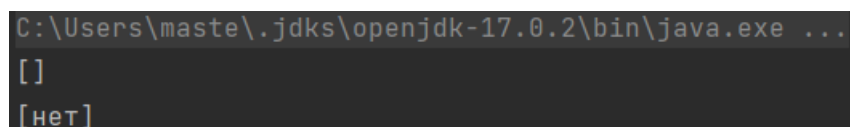
Рисунок 4. Результат выполнения программы

Задание: Вариант 3

3. Найти такое слово в первом предложении, которого нет ни в одном из остальных предложений.

Ход работы: Код программы файла Task_3_3

```
import java.util.ArrayList;  
  
public class Task_3_3 {  
    public static void main(String[] arg) {  
        String str = "Я нет." +  
            "Я тут." + " Всем пока.";   
  
        String start = str.substring(0, str.indexOf(".")); //первое предложение  
        String end = str.substring(str.indexOf(".") + 1); //остальная часть без первого предложения  
  
        String[] word = start.replaceAll(",|:|\\(|\\)", "").split(" ");  
        //разбиваем первое предложение на слова  
  
        ArrayList<String> yesWord = new ArrayList<>(); //Хранилище для слов которые есть в тексте  
        ArrayList<String> noWord = new ArrayList<>(); // Хранилище для слов которые не встречаются в тексте  
  
        for (String s : word) {  
            if (end.contains(s)) { //если есть слово из первого предложения в остальном тексте  
                yesWord.add(s);  
            }  
            else {  
                noWord.add(s);  
            }  
            System.out.println(noWord);  
        }  
    }  
}
```



```
C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...  
[ ]  
[нет]
```

Рисунок 5. Результат выполнения программы

4. Во всех вопросительных предложениях текста найти и напечатать без повторений слова заданной длины.

Код программы файла Task_3_4

```
import java.util.Scanner;
import java.util.StringTokenizer;
public class Task_3_4 {
    public static void main(String[] arg) {
        String Text = "Я нет мы что здесь?" +
            "Я тут." + " Всем пока вот как?";
        Scanner sc = new Scanner(System.in);
        int len;

        System.out.print("Введите длину слов: ");
        len = sc.nextInt();

        // делим текст на предложения и разделители
        StringTokenizer stringTokenizer = new
StringTokenizer(Text, "\n:!.?", true);
        String last = "";
        String current = "";
        String[] dic = new String[0];
        String[] dicUnique = new String[0];

        while (stringTokenizer.hasMoreTokens())
        {
            // сохраняем очередное предложение во временной переменной
            last = current;
            // получаем очередное "предложение"
            current = stringTokenizer.nextToken().trim();

            // если полученное предложение является токеном "?"
            if (current.equals("?")==true)
            {
                // добавляем предыдущее предложение в массив предложений
                String[] dic2 = new String[dic.length + 1];
                System.arraycopy(dic, 0, dic2, 0, dic.length);
                dic2[dic.length] = last+current;
                dic = new String[dic2.length];
                System.arraycopy(dic2, 0, dic, 0, dic2.length);
            }
        }
        // вывод результатов отбора вопросительных предложений
        if (dic.length>0)
        {
            System.out.println("\nСписок вопросительных предложений:");
            for (int i = 0; i < dic.length; i++)
            {
                System.out.println(dic[i]);
            }

            // разбиение вопросительных предложений на слова и выборка
            уникальных слов указанной длины

            for (int i = 0; i < dic.length; i++)
            {
                StringTokenizer stringTokenizer2 = new
StringTokenizer(dic[i], " \n\t,.;!?'\"-\\");
                while (stringTokenizer2.hasMoreTokens())
                {
```

```

// получаем очередное слово
String tok = stringTokenizer2.nextToken().trim();
if (tok.length()==len)
{
    // ищем слово среди уже существующих в массиве
    boolean b = false;
    for (int j = 0; j < dicUnique.length; j++)
    {
        String string = dicUnique[j];
        // если слово найдено
        if (string.equalsIgnoreCase(tok)==true)
        {
            b = true;
            break;
        }
    }
    // если слово не было найдено в списке слов нужной
    if (b==false)
    {
        // добавляем слово в массив слов нужной длины
        String[] dic2 = new String[dicUnique.length +
1];
        System.arraycopy(dicUnique, 0, dic2, 0,
dicUnique.length);
        dic2[dicUnique.length] = tok;
        dicUnique = new String[dic2.length];
        System.arraycopy(dic2, 0, dicUnique, 0,
dic2.length);
    }
}
}
}
if (dicUnique.length>0)
{
    System.out.printf("\nСписок уникальных слов указанной
длины (%d):\n",len);
    for (int i = 0; i < dicUnique.length; i++)
        System.out.println(dicUnique[i]);
}
else System.out.printf("\nВ вопросительных предложениях не
найден ни одного слово указанной длины (%d)!\n", len);
}
else System.out.printf("\nНе найдено ни одного вопросительного
предложения!\n", len);
}
}

```

```

C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...
Введите длину слов: 3

Список вопросительных предложений:
Я нет мы что здесь?
Всем пока вот как?

Список уникальных слов указанной длины (3):
нет
что
вот
как

```

Рисунок 6. Результат выполнения программы

Задание: Вариант 4

3. Все слова текста рассортировать в порядке убывания их длин, при этом все слова одинаковой длины рассортировать в порядке возрастания в них количества гласных букв.

Ход работы: Код программы файла Task_4_3

```
import java.util.*;
public class Task_4_3 {
public static void main(String[] args){
    String text = ""
        Равным образом сложившаяся структура организации обеспечивает
широкому кругу специалистов участие\с
        в формировании новых предложений. Практический опыт
показывает, что реализация намеченного плана развития обеспечивает широкому\s
        кругу специалистов участие в формировании системы обучения
кадров, соответствующей насущным потребностям?""";

    Comparator<String> comparator = new Comparator<String>() {
        @Override
        public int compare(String o1, String o2) {
            String alpha = "аеёиоуэя";
            int buf_1, buf_2;
            if (o1.length() > o2.length()){
                return -1;
            } else if (o1.length() < o2.length()){
                return 1;
            } else{
                buf_1 = buf_2 = 0;
                for (char sym : o1.toCharArray()){
                    if (alpha.indexOf(sym) != -1){
                        buf_1++;
                    }
                }
                for (char sym : o2.toCharArray()){
                    if (alpha.indexOf(sym) != -1){
                        buf_2++;
                    }
                }
                if (buf_1 > buf_2){
                    return 1;
                } else return -1;
            }
        }
    };

    String[] words =
text.toLowerCase(Locale.ROOT).replaceAll("[,.\n]", "").split("\\s");
    List<String> words_list = new
ArrayList<String>(Arrays.asList(words));
    words_list.sort(comparator);
    for (String word : words_list){
        System.out.println(word);
    }
}
```



```

C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...
соответствующей
потребностям?
практический
специалистов
специалистов
формировании
обеспечивает
формировании
обеспечивает
предложений
намеченного
сложившаяся
организации
показывает
реализация
структура
насущным
широкому
развития
широкому
обучения

```

Рисунок 7. Результат выполнения программы

4. В тексте исключить подстроку максимальной длины, начинающуюся и заканчивающуюся заданными символами.

Ход работы: Код программы файла Task_4_4

```

import java.util.regex.Pattern;
import java.util.regex.Matcher;
import java.util.*;
class Task_4_4 {
    public static void main(String[] args) {
        String text = "пая паая нет на пальме, но есть на другой
и кричит пая.";
        String[] wordArray = text.split("[\\s,.:!?]+");
        Pattern pattern = Pattern.compile("^([п].*[я])$");
        String maxString = "";
        String newStr = "";
        for (String word : wordArray) {
            if (word.length() > maxString.length()) {
                maxString = word;
                Matcher matcher = pattern.matcher(maxString);
                if (matcher.find() ) {
                    newStr = text.replaceAll(maxString, "");
                }
            }
        }
    }
}

```

```
        System.out.println(newStr);  
    }  
}
```

```
C:\Users\maste\.jdk\openjdk-17.0.2\bin\java.exe ...  
пая нет на пальме, но есть на другой и кричит пая.
```

Рисунок 8. Результат выполнения программы

Вывод: лабораторная работа была выполнена в соответствии с заданием и полученные верные результаты работ программ