



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, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ 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

Название: Строки и регулярные выражения

Дисциплина: Языки программирования для работы с большими
данными

Вариант: 2

Студент

ИУ6-22М

(Группа)

(Подпись, дата)

Н.А. Аскерова

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

П.В. Степанов

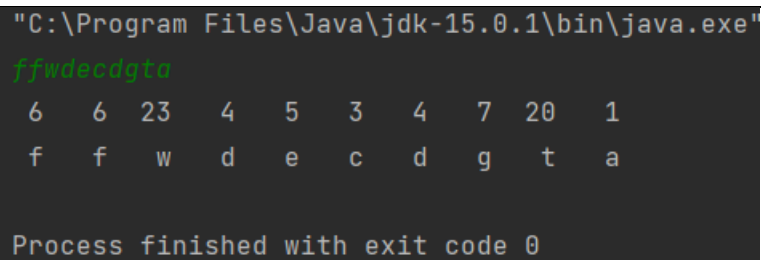
(И.О. Фамилия)

Вариант 1

2. В 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 1 – Код программы

```
import java.io.*;
import java.nio.charset.StandardCharsets;
import java.util.Scanner;
class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String text = scanner.nextLine().toLowerCase();
        int n = text.length();
        int[] numbers = new int[n];
        for (int i = 0; i < n; i++) {
            char c = text.charAt(i);
            if (c >= 'a' && c <= 'z') {
                numbers[i] = c - 'a' + 1;
            } else {
                numbers[i] = 0;
            }
        }
        for (int i = 0; i < n; i++) {
            System.out.printf("%2d ", numbers[i]);
        }
        System.out.println();
        for (int i = 0; i < n; i++) {
            System.out.printf("%2c ", text.charAt(i));
        }
        System.out.println();
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"
ffwdecgta
 6  6 23  4  5  3  4  7 20  1
f  f  w  d  e  c  d  g  t  a

Process finished with exit code 0
```

Рисунок 1 – Результат работы программы

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

Листинг 2 – Код программы

```
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        String text = "PAJKJ P A PPIHJH PAA";
        String correctedText = text.replaceAll("(?<=P)A(?!=\\p{L})", "O");
        System.out.println(correctedText);
    }
}
```

```
"C:\Program Files\Java\jdk-15.0.1\bin\j
POJKJ P A PPIHJH POA

Process finished with exit code 0
```

Рисунок 2 – Результат работы программы

Вариант 2

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

Листинг 3 – Код программы

```
import java.util.HashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) {
        String text = "hello dear friend hello ";
        Map<String, Integer> wordFreq = new HashMap<>();
        String[] words = text.split("\\s+");

        for (String word : words) {
            if (wordFreq.containsKey(word)) {
                wordFreq.put(word, wordFreq.get(word) + 1);
            } else {
                wordFreq.put(word, 1);
            }
        }

        for (Map.Entry<String, Integer> entry : wordFreq.entrySet()) {
            System.out.println(entry.getKey() + " : " + entry.getValue());
        }
    }
}
```

```
"C:\Program Files\Java\jdk-15.0.1\bin\j
friend : 1
hello : 2
dear : 1

Process finished with exit code 0
```

Рисунок 3 – Результат работы программы

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

Листинг 4 – Код программы

```
import java.util.HashMap;
import java.util.Map;
import java.util.stream.Collectors;

public class Main {
    public static void main(String[] args) {
        String text = "In the text, find and print the 5 characters (and their number) that occur most often.";
```

```

Map<Character, Integer> charFreq = new HashMap<>();

for (char c : text.toCharArray()) {
    if (charFreq.containsKey(c)) {
        charFreq.put(c, charFreq.get(c) + 1);
    } else {
        charFreq.put(c, 1);
    }
}

String top5Chars = charFreq.entrySet() //получение потока кодов символов (IntStream) из строки
    .stream()
    .sorted(Map.Entry.<Character, Integer>comparingByValue().reversed()) // сортировка (по убыванию
количества вхождений)
    .limit(5) //ограничение количества результирующих записей
    .map(entry -> entry.getKey() + " : " + entry.getValue())
    .collect(Collectors.joining(", ")); //подсчет количества вхождений каждого символа

System.out.println("Top 5 most frequently occurring characters: " + top5Chars);
}
}

```

```

"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBra
Top 5 most frequently occurring characters:   : 15, t : 11, e : 7, n : 7, r : 6

Process finished with exit code 0

```

Рисунок 4 – Результат работы программы

Вариант 3

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

Листинг 5 – Код программы

```

import java.util.HashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) {
        String text = "cat dog dog hello hi. "
            + "print cat cat cat lock cat."
            + "hello bye dog.";
        String[] sentences = text.split("(?<=[?!])\\s+");
        int maxSentencesWithIdenticalWords = 0;

        for (String sentence : sentences) {
            String[] words = sentence.split("\\s+");
            Set<String> uniqueWords = new HashSet<>();
            int numIdenticalWords = 0;

            for (String word : words) {
                if (!uniqueWords.add(word)) {
                    numIdenticalWords++;
                }
            }

            if (numIdenticalWords > 0 && numIdenticalWords > maxSentencesWithIdenticalWords) {
                maxSentencesWithIdenticalWords = numIdenticalWords;
            }
        }
    }
}

```

```

    }
}

System.out.println("The largest number of sentences with identical words is: " +
maxSentencesWithIdenticalWords);
}
}

```

```

C:\Program Files\Java\jdk-15.0.1\bin\java.exe -javaagent:C:\Program Files\JetBrains\IntelliJ I
The largest number of sentences with identical words is: 2

Process finished with exit code 0

```

Рисунок 5 – Результат работы программы

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

Листинг 6 – Код программы

```

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

public class Main {
    public static void main(String[] args) {
        String text = "cat dog print lock. "
            + "cat dog hello. "
            + "print dog light.";
        String[] sentences = text.split("(?<=[?!])\\s+");
        String[] wordsInFirstSentence = sentences[0].split("\\s+");
        Set<String> uniqueWordsInFirstSentence = new HashSet<>(Arrays.asList(wordsInFirstSentence));
        Set<String> wordsInOtherSentences = new HashSet<>();

        for (int i = 1; i < sentences.length; i++) {
            String[] words = sentences[i].split("\\s+");
            wordsInOtherSentences.addAll(Arrays.asList(words));
        }

        uniqueWordsInFirstSentence.removeAll(wordsInOtherSentences);

        if (!uniqueWordsInFirstSentence.isEmpty()) {
            System.out.println("A word in the first sentence that is not present in any other sentence is: " +
uniqueWordsInFirstSentence.iterator().next());
        } else {
            System.out.println("No unique words found in the first sentence.");
        }
    }
}

```

```

"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ I
A word in the first sentence that is not present in any other sentence is: lock.

Process finished with exit code 0

```

Рисунок 6 – Результат работы программы

Вариант 4

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 7 – Код программы

```
import java.util.*;

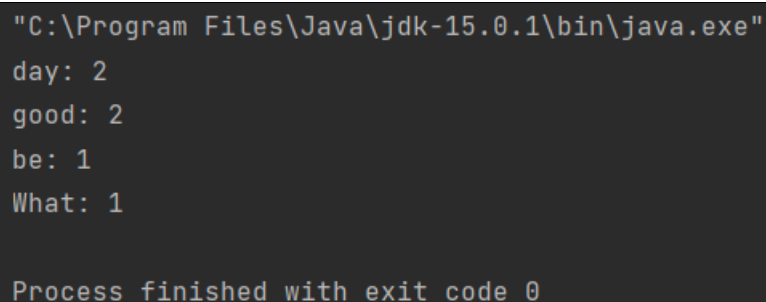
public class Main {
    public static void main(String[] args) {
        String text = "What a good day I'm having today!\n" +
            "A good day to be free.";
        List<String> words = Arrays.asList("day", "good", "be", "What");

        Map<String, Integer> frequencyMap = new HashMap<>();
        for (String word : words) {
            frequencyMap.put(word, 0);
        }

        String[] sentences = text.split("[.?!]\\s*");
        for (String sentence : sentences) {
            String[] sentenceWords = sentence.split("\\s+");
            for (String word : sentenceWords) {
                if (frequencyMap.containsKey(word)) {
                    frequencyMap.put(word, frequencyMap.get(word) + 1);
                }
            }
        }

        List<Map.Entry<String, Integer>> sortedList = new ArrayList<>(frequencyMap.entrySet());
        sortedList.sort((a, b) -> b.getValue().compareTo(a.getValue()));

        for (Map.Entry<String, Integer> entry : sortedList) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```



```
"C:\Program Files\Java\jdk-15.0.1\bin\java.exe"
day: 2
good: 2
be: 1
What: 1

Process finished with exit code 0
```

Рисунок 7 – Результат работы программы

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.

Листинг 8 – Код программы

```
import java.util.*;

public class Main {
```

```

public static void main(String[] args) {
    String text = "letters in for aim accuracy file them";
    String[] words = text.split("\\s+");

    Arrays.sort(words, (a, b) -> {
        int lengthCompare = Integer.compare(b.length(), a.length());
        if (lengthCompare != 0) {
            return lengthCompare;
        } else {
            int vowelCompare = Integer.compare(countVowels(a), countVowels(b));
            return vowelCompare;
        }
    });

    for (String word : words) {
        System.out.println(word);
    }
}

private static int countVowels(String word) {
    int count = 0;
    for (char c : word.toCharArray()) {
        if ("AEIOUaeiou".indexOf(c) != -1) {
            count++;
        }
    }
    return count;
}
}

```

```

"C:\Program Files\Java\jdk-15.0.1\bin\java.exe" "-j
accuracy
letters
them
file
for
aim
in

Process finished with exit code 0

```

Рисунок 8 – Результат работы программы

Вывод: приобретен навык работы со строками и регулярными выражениями.

Ссылка на репозиторий с программами: <https://github.com/nargi3/BigData>