



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 № 6

**Название лабораторной работы:** Строки. Регулярные выражения

**Дисциплина:** Языки программирования для работы с большими данными

Студент гр. ИУ6-22М

\_\_\_\_\_  
(Подпись, дата)

**И.А. Горшков**  
(И.О. Фамилия)

Преподаватель

\_\_\_\_\_  
(Подпись, дата)

**П.В. Степанов**  
(И.О. Фамилия)

Москва, 2023

## **Введение**

Целью лабораторной работы является приобретение навыков работы со строками и регулярными выражениями на языке программирования Java.

## Практическая часть

### Задание 1

6. После каждого слова текста, заканчивающегося заданной подстрокой, вставить указанное слово.

Код написанной программы представлен в листинге 1.

Листинг 1 – Программа для первого задания

```
package com.java.lab;

// После каждого слова текста, заканчивающегося заданной подстрокой,
// вставить указанное слово.
public class Main {

    public static void main(String[] args) {
        var text = "They are conducting more tests.";
        System.out.println(makeStringWithWordAfterSubstring(text,
"more", "lord"));
    }

    public static String makeStringWithWordAfterSubstring(String text,
String substring, String word) {
        var builder = new StringBuilder();
        var words = text.split(" ");

        for (String s : words) {
            builder.append(s);
            if (s.endsWith(substring)) {
                builder.append(" ");
                builder.append(word);
            }
            builder.append(" ");
        }

        return builder.toString();
    }
}
```

Результат выполнения программы показан на рисунке 1.

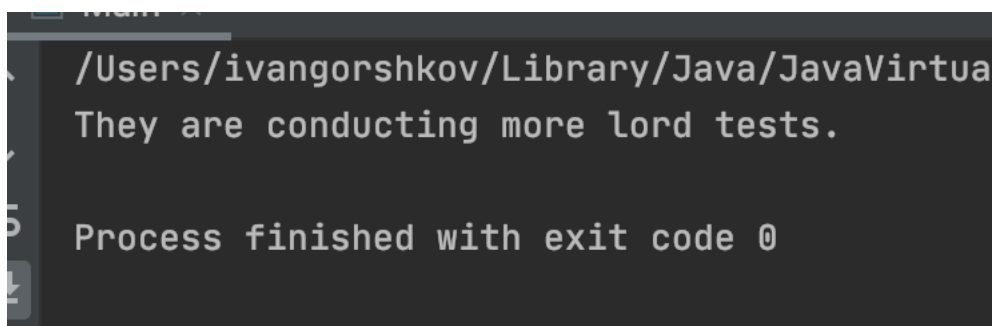


Рисунок 1 – Результат выполнения программы

## Задание 2

В зависимости от признака (0 или 1) в каждой строке текста удалить указанный символ везде, где он встречается, или вставить его после k-го символа.

Код написанной программы представлен в листинге 2.

### Листинг 2 – Программа для второго задания

```
package com.java.lab;
import java.util.Scanner;

// В зависимости от признака (0 или 1) в каждой строке текста удалить
// указанный символ везде, где он встречается, или вставить его после k-
// го символа.
public class Main {
    public static void main(String[] args) {
        var scanner = new Scanner(System.in);

        var text = "They are conducting more tests.";

        System.out.println("symbol:");
        var symbol = scanner.nextLine().charAt(0);

        System.out.println("flag:");
        var flag = scanner.nextInt();

        System.out.println("k:");
        var k = scanner.nextInt();

        var lines = text.split("\n");
        for (var line : lines) {
            if (flag == 0) {
                line = line.replaceAll(Character.toString(symbol),
""");
                System.out.println(line);
            } else {
                StringBuilder builder = new StringBuilder(line);
                builder.insert(k, symbol);
                System.out.println(builder.toString());
            }
        }
    }
}
```

Результат выполнения программы показан на рисунке 2.

```
/Users/ivangorshkov/Library/Java/JavaVirtualMachines/
symbol:
r
flag:
1
k:
2
They are conducting more tests.

Process finished with exit code 0
```

Рисунок 2 – Результат выполнения программы

### Задание 3

Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

Код написанной программы представлен в листинге 3.

#### Листинг 3 – Программа для третьего задания

```
package com.java.lab;
import java.util.HashSet;

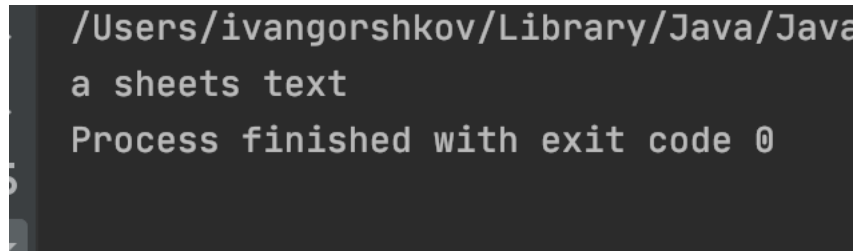
// Напечатать без повторения слова текста, у которых первая и
// последняя буквы совпадают.
public class Main {
    public static void main(String[] args) {
        var text = "Lorem Ipsum is simply dummy text of the printing
and typesetting industry. Lorem Ipsum has been the industry's standard
dummy text ever since the 1500s, when an unknown printer took a galley
of type and scrambled it to make a type specimen book. It has survived
not only five centuries, but also the leap into electronic
typesetting, remaining essentially unchanged. It was popularised in
the 1960s with the release of Letraset sheets containing Lorem Ipsum
passages, and more recently with desktop publishing software like
Aldus PageMaker including versions of Lorem Ipsum.";
        var set = new HashSet<>();
        var words = text.split(" ");

        for (var word : words) {
            if (word.charAt(0) == word.charAt(word.length() - 1)) {
                set.add(word);
            }
        }

        for (var word : set) {
            System.out.print(word + " ");
        }
    }
}
```

```
}  
}  
}
```

Результат выполнения программы показан на рисунке 3.



```
/Users/ivangorshkov/Library/Java/Java  
a sheets text  
Process finished with exit code 0
```

Рисунок 3 – Результат выполнения программы

#### Задание 4

В тексте найти и напечатать все слова максимальной и все слова минимальной длины.

Код написанной программы представлен в листинге 4.

Листинг 4 – Программа для четвертого задания

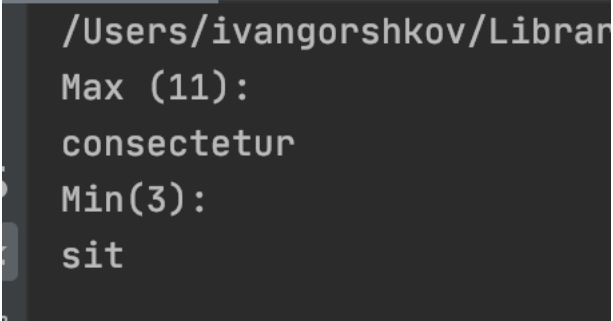
```
package com.java.lab;  
  
// В тексте найти и напечатать все слова максимальной и все слова  
минимальной длины.  
public class Main {  
  
    public static void main(String[] args) {  
        var text = "Lorem ipsum dolor sit amet, consectetur adipiscing  
elit. Fusce luctus.";  
  
        var words = text.split(" ");  
  
        var maxLength = 0;  
        var minLength = words[0].length();  
  
        for (var word : words) {  
            if (word.length() > maxLength) {  
                maxLength = word.length();  
            }  
            if (word.length() < minLength) {  
                minLength = word.length();  
            }  
        }  
  
        System.out.println("Max (" + maxLength + "):");  
        for (var word : words) {  
            if (word.length() == maxLength) {  
                System.out.println(word);  
            }  
        }  
  
        System.out.println("Min(" + minLength + "):");  
        for (var word : words) {
```

```

        if (word.length() == minLength) {
            System.out.println(word);
        }
    }
}

```

Результат выполнения программы показан на рисунке 4.



```

/Users/ivangorshkov/Library
Max (11):
consectetur
Min(3):
sit

```

Рисунок 4 – Результат выполнения программы

### Задание 5

В предложении из  $n$  слов первое слово поставить на место второго, второе – на место третьего, и т.д.,  $(n-1)$ -е слово – на место  $n$ -го,  $n$ -е слово поставить на место первого. В исходном и преобразованном предложениях между словами должны быть или один пробел, или знак препинания и один пробел.

Код написанной программы представлен в листинге 5.

### Листинг 5 – Программа для пятого задания

```

package com.java.lab;

// В предложении из n слов первое слово поставить на место второго,
// второе – на место третьего, и т.д., (n-1)-е слово – на место n-го, n-е
// слово поставить на место первого. В исходном и преобразованном
// предложениях между словами должны быть или один пробел, или знак
// препинания и один пробел.
public class Main {
    public static void main(String[] args) {
        var sentence = "Lorem ipsum dolor sit amet, consectetur
adipiscing elit. Fusce luctus.";
        System.out.println(sentence);

        var words = sentence.split("\\s+");
        var offsetWords = new String[words.length];

        for (int i = 0; i < words.length; i++) {
            int newIndex = i == words.length - 1 ? 0 : i + 1;
            offsetWords[newIndex] = words[i];
        }

        var offsetSentence = new StringBuilder();
    }
}

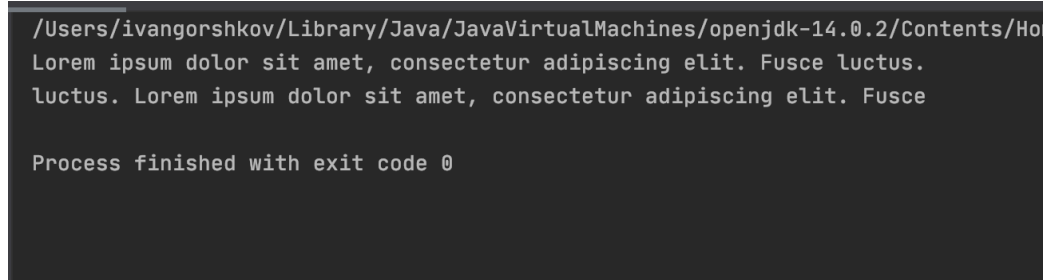
```

```

        for (String reversedWord : offsetWords) {
            offsetSentence.append(reversedWord).append(" ");
        }
        System.out.println(offsetSentence);
    }
}

```

Результат выполнения программы показан на рисунке 5.



```

/Users/ivangorshkov/Library/Java/JavaVirtualMachines/openjdk-14.0.2/Contents/Home
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce luctus.
luctus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce

Process finished with exit code 0

```

Рисунок 5 – Результат выполнения программы

### Задание 6

Текст шифруется по следующему правилу: из исходного текста выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы, затем 2, 5, 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д. Зашифровать заданный текст.

Код написанной программы представлен в листинге 6.

Листинг 6 – Программа для шестого задания

```

package com.java.lab;

// Текст шифруется по следующему правилу: из исходного текста
// выбирается 1, 4, 7, 10-й и т.д. (до конца текста) символы, затем 2, 5,
// 8, 11-й и т.д. (до конца текста) символы, затем 3, 6, 9, 12-й и т.д.
// Зашифровать заданный текст.
public class Main {

    public static void main(String[] args) {
        var text = "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Fusce luctus.";
        System.out.println(text);
        var encryptedText = encrypt(text);
        System.out.println(encryptedText);
    }

    public static String encrypt(String text) {
        var result = new StringBuilder();
        for (int i = 0; i < text.length(); i += 3) {
            result.append(text.charAt(i));
        }
        for (int i = 1; i < text.length(); i += 3) {
            result.append(text.charAt(i));
        }
        for (int i = 2; i < text.length(); i += 3) {

```

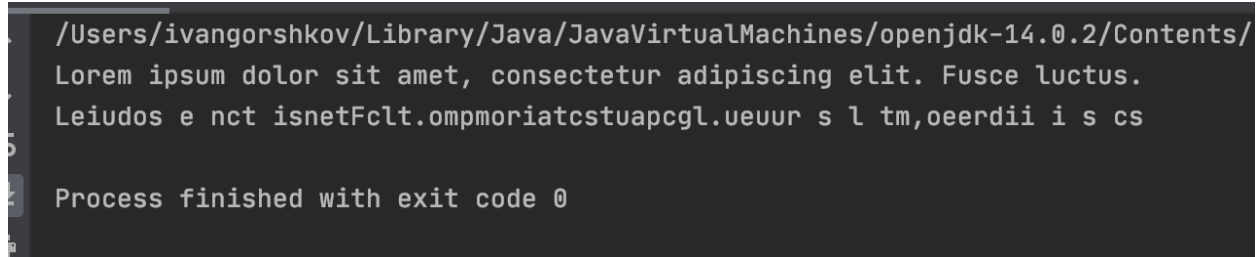


```

        result.append(text.charAt(i));
    }
    return result.toString();
}
}

```

Результат выполнения программы показан на рисунке 6.



```

/Users/ivangorshkov/Library/Java/JavaVirtualMachines/openjdk-14.0.2/Contents/
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce luctus.
Leiudos e nct isnetFclt.ompmoriatcstuapcgl.ueuur s l tm,oeerdii i s cs
2
Process finished with exit code 0

```

Рисунок 6 – Результат выполнения программы

### Задание 7

Вывести в заданном тексте все слова, расположив их в алфавитном порядке.

Код написанной программы представлен в листинге 7.

Листинг 7 – Программа для седьмого задания

```

package com.java.lab;

import java.util.Arrays;

// Вывести в заданном тексте все слова, расположив их в алфавитном
// порядке.
public class Main {

    public static void main(String[] args) {
        var text = "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nam venenatis eleifend consequat. Praesent bibendum euismod
quam, a rutrum ex consectetur.";
        var words = text.toLowerCase().split("[\\p{Punct}\\s]+");
        Arrays.sort(words);
        for (var word : words) {
            System.out.println(word);
        }
    }
}

```

Результат выполнения программы показан на рисунке 7.

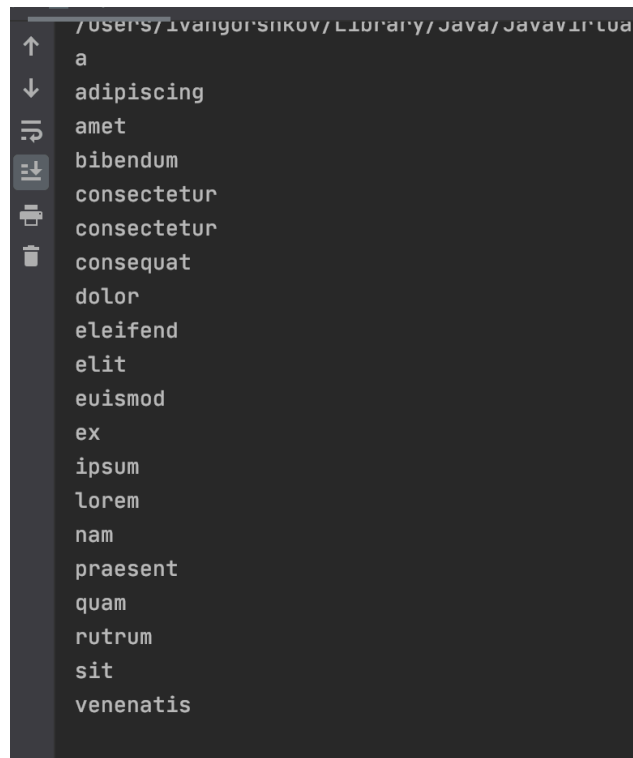


Рисунок 7 – Результат выполнения программы

### Задание 8

Подсчитать, сколько слов в заданном тексте начинается с прописной буквы.

Код написанной программы представлен в листинге 8.

### Листинг 8 – Программа для четвертого задания

```
package com.java.lab;

// Подсчитать, сколько слов в заданном тексте начинается с прописной
буквы.
public class Main {

    public static void main(String[] args) {
        var text = "Lorem ipsum dolor sit amet, consectetur adipiscing
elit. Nam venenatis eleifend consequat. Praesent bibendum euismod
quam, a rutrum ex consectetur.";
        System.out.println(text);
        System.out.println(countCapitalizedWords(text));
    }

    public static int countCapitalizedWords(String text) {
        var words = text.split("\\s+");
        var count = 0;
        for (String word : words) {
            if (word.matches("[A-ЯА-З][a-яа-з]*")) {
                count++;
            }
        }
        return count;
    }
}
```

```
}  
}
```

Результат выполнения программы показан на рисунке 8.

```
/Users/ivangorshkov/Library/Java/JavaVirtualMachines/openjdk-14.0.2/Contents/Home/bin/java --enable-preview -javaagent:/Applications/IntelliJ IDEA.app/Contents/  
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam venenatis eleifend consequat. Praesent bibendum euismod quam, a rutrum ex consectetur.  
3  
  
Process finished with exit code 0
```

Рисунок 8 – Результат выполнения программы

**Вывод:** В результате выполнения лабораторной работы были приобретены навыки работы со строками и регулярными выражениями на языке программирования Java.

Репозитории по ссылке <https://github.com/IvanGorshkov/Java-IU6-12M>.