

Лабораторная работа 09. Строки

1. Ввести с клавиатуры одну строку

- а) вывести количество символов 'S' и 's' (по отдельности и общее их количество)
- б) вывести количество цифр в этой строке.
- в) вывести количество заглавных букв в строке
- д) используя String.format() вывести все результаты для пунктов а),б),в) в следующей форме:

```
Исходная строка = "123abc-S-s-+++SSSaaaAAAssss5678",  
количество S = 4,  
количество s = 5,  
количество s и S = 9,  
количество цифр = 7,  
количество заглавных букв = 7
```

2. Ввести с клавиатуры одну строку, строка состоит из слов, разделенных пробелами.

Сделать из строки массив, содержащих отдельные слова этой строки (использовать .split())

Вывести количество полученных слов.

Вывести количество слов "red" (учитывать все варианты прописных и строчных букв).

Каждый третий элемент массива заменить на слово "green".

Сформировать и вывести новую строку, в которой все слова из массива разделены запятой и пробелом (" , ")

3. Ввести с клавиатуры строку.

Сформировать и вывести новую строку, в которой заменить все буквы на заглавные

- а) Сформировать и вывести новую строку, в которой заменить все "min" на "max". Использовать replace
- б) Сформировать и вывести новую строку, в которой заменить все вхождения символа '1' на '5', 'a' на 'A', 'f' на 'F', 'g' на 'G'. Использовать обработку отдельных символов в цикле
- в) Сформировать и вывести новую строку, в которой удалить все подстроки "ku-ku"
- д) если исходная строка включала подстроку "-abc-",
то вывести только индекс символа с которого начинается это вхождение.
если нет, то вставить ""-abc-" в середину исходной строки

<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>

<https://metanit.com/java/tutorial/7.1.php>

Про printf() и String.format() <http://study-java.ru/uroki-java/formatirovanie-chisel-i-texta-v-java/>

Примеры

Использование класса Character

```
char sp = ' ';
char ch = '!';
char a = '*';
char b = 'W';
char c = 'Ю';
char d = 65; // 'A'
char e = 0xA66C; // 'ᄐ'
char f = '\uA64A'; // 'ᄒ'

int x = b + 2; // Y
//short y = b - 5; // ошибка b-5 мана int(4 байта),
short y = (short)(b - 6); // 87 - 6 = 81
char r = (char) y; // 81 = 'Q'

char zero = 48;
char eight = '8';
char five = (char)(eight - 3); // = '5'
char two = (char)(zero + 2); // = '2'

// управляющие символы
// '\n' '\t'

Character spp = ' ';
Character cch = '!';
Character aa = '*';
Character bb = 'W';
Character cc = 'Ю';
Character ff = '5';

if ( Character.isDigit(ff) ) { // true //...
}

if ( Character.isAlphabetic(bb) ) { // true //...
}

if ( Character.isLetter(cc) ) { // true //...
}
// System.out.println('\u2164'); // римская цифра V - isAlphabetic, ! isLetter

if ( Character.isSpaceChar(spp) ) { // true //...
}

if ( Character.isUpperCase('F') ) { // true
//...
}

b = Character.toLowerCase('R'); // 'r'
b = Character.toUpperCase('r'); // 'R'

// b = 0x1F680; // ошибка, 0x1F680 = 128640 не умещается в 2 байта
char bHigh = 0xd83d;
char bLow = 0xde80;
System.out.println(bHigh + "" + bLow);

int code = Character.toCodePoint(bHigh, bLow);
char[] chars = Character.toChars(code);
System.out.println(chars.length);
for (var ccc: chars) {
    System.out.print(ccc);
}
}
```

Использование класса String

```
char[] cc = {'a', 'b', 'c', 'd'}; // символьный массив != строка
// String str;
// String str = null;
// System.out.println(str); // неинициализированную строку использовать нельзя

String s = "Корова";
String ss = "Корова\tМурка и \nМатроскин";
String sss = "\u041a\u043e\u0440\u043e\u0432\u0430"; // "Корова"
String ssss = new String(new char[]{'a', 'b', 'c', 'd'});

// размещение в памяти: строковый пул + immutable
String s1 = "Вася";
String s2 = new String("Вася");
String s3 = "Вася";

//сравнение ссылок на строки
System.out.println( s1 == s2 ); // false
System.out.println( s1 == s3 ); // true
System.out.println( s2 == s3 ); // false
System.out.println("-----");

// сравнение самих строк в лексикографическом порядке
System.out.println( s1.equals(s2) ); // true
System.out.println( s1.equals(s3) ); // true
System.out.println( s2.equals(s3) ); // true

if ( "Вася".equals(s3) ) { } // так лучше
if ( s3.equals("Вася") ) { } // чем так

System.out.println("-----");

// формирование строки статическими методами класса String
String str = String.format("строка в которую %s добавили %d других %.3f данных",
                           "где-то", 1000, Math.PI);

System.out.println(str);
System.out.println("-----");

str = String.join(" - ", "Вася", "Петя", "Коля");
System.out.println(str);

String[] names = new String[]{"Вася", "Петя", "Коля"};
str = String.join(" - ", names);
System.out.println(str);

System.out.println("-----");

// конкатенация
s1 = "Вася";
System.out.println(s1 + " + Дуся" + " = " + "♥"); // терпимо
System.out.println(s1.concat(" + Дуся").concat(" = ").concat("♥"));

// конкатенация и повторяющиеся символы
s1 = "";
for (int i = 0; i < 100; i++) { // плоховато
    s1 = s1 + "♥";
}
System.out.println(s1);

s1 = "♥".repeat(100); // только для java 11 и выше
System.out.println(s1);

// создание строки на основе массива символов
char[] c = {'a', 'b', 'c', 'd', 'e', 'f', 'g'};
String a = new String(c);
System.out.println(a);

a = new String(c, 2, 3);
```

```
System.out.println(a);
```

```
c = new char[100];  
Arrays.fill(c, '♥');  
String h1 = new String(c);  
System.out.println(h1);
```

```
// использование экземплярных методов класса String
```

```
String h2 = new String(new char[100]).replace('\0', '♥');
```

```
//----- методы класса String
```

```
// !!! эти методы возвращают новую, преобразованную строку, но НЕ ИЗМЕНЯЮТ исходную
```

```
s1 = "Здесь был Петя ";  
System.out.println(s1.toUpperCase());
```

```
s1 = s1.toLowerCase();
```

```
s1 = s1.replace("e", "Э"); // замена
```

```
s1 = s1.trim(); // убрать пробелы по краям
```

```
String[] words = s1.split(" "); // разделить строку на отдельные строки  
System.out.println(Arrays.toString(words));
```

```
s1.substring(9, 13); // подстрока
```

```
s1.indexOf("был"); // место входа подстроки или -1
```

```
s1.compareTo(s2); // сравнение в лексикографическом порядке, если совпали, то 0; иначе +/-
```

```
//-----
```

```
// посимвольная обработка - только чтение
```

```
System.out.println("5 - это цифра " + Character.isDigit('5'));
```

```
int len = s1.length();  
for (int i = 0; i < len; i++) {  
    if (s1.charAt(i) == 'Z') {  
        System.out.println("Да, это буква ");  
    }  
}
```

```
// то же через foreach - надо преобразовать в итерируемый (Iterable) тип
```

```
for (var ch : s1.toCharArray()) {  
    if (ch == 'Z') {  
        System.out.println("Да, это буква ");  
    }  
}
```