Лабораторная работа № 7

РАБОТА СО СТРОКАМИ

СТРОКА - последовательность символов, заключенная в кавычки. В JAVA строка представлена объектным типом String.

Объявление строки:

```
String <название переменной> = new String(<строка>);
```

Оператор создания объекта-строки `new String(<строка>)` можно заменить указанием новой строки в кавычках:

```
String <название переменной> = <строка>;
```

Примеры:

Объявление переменной s и присвоение ей строки “Hello”:

```
String s1 = new String(“Hello”);
```

или

```
String s1 = “Hello”;
```

Объявление пустой строки “”:

```
String s2 = new String();
```

или

```
String s2 = “”;
```

Кроме типа String в JAVA присутствует тип char, который используется для представления одного символа. Тип char является примитивным типом. Значение переменной типа `char` задается в апострофах, например:

```
char c = ‘a’;
```

Если имеется массив переменных типа `char`, можно на основе этого массива создать новую строку, например:

```
char c[] = new char[] {‘h’, ‘e’, ‘l’, ‘l’, ‘o’};
String s = new String(c);
```

Обработка строк

Выражения, в которых операндами служат строковые данные, называются строковыми. Так как строка в JAVA представлена в виде объектного типа, основные методы по обработке строк содержатся «внутри» самого объекта-строки и используют его значение в качестве параметра. Например, вызов метода `.length()` у строковой переменной вернет длину (количество символов) строки, с которой связана данная переменная.

```
String s = new String(“Hello”);
System.out.println(“Длина строки s равна” + s.length());
```

Некоторые методы строковых объектов приведены ниже:

`int length()` – возвращает длину строки (`int`)

`char charAt(int <индекс>)` – возвращает символ строки (`char`) по указанному индексу. `<индекс>` - от 0 до length()–1

`char[] toCharArray()` – возвращает строку в виде массива символов (`char[]`)

`boolean equals(Object <строка1>)` – сравнивает данную строку со `строкой1`. Возвращает `true` если строки совпадают, или `false` (`boolean`) – в обратном случае.

`boolean equalsIgnoreCase(String <строка1>)` – сравнивает данную строку со `строкой1` без учета регистра символов. Возвращает `true` если строки совпадают, или `false` (`boolean`) – в обратном случае.

`int indexOf(String <строка1>)` – возвращает позицию (индекс) первого вхождения `строки1` в данную строку. Если не найдено – возвращает –1.

`int indexOf(String <строка1>, int <начальный_индекс>)` – возвращает позицию (индекс) первого вхождения `строки1` в данную строку. Поиск вхождения осуществляется с указанного начального индекса. Если не найдено – возвращает –1.

`int lastIndexOf(String <строка1>)` – возвращает позицию (индекс) последнего вхождения `строки1` в данную строку. Если не найдено – возвращает –1.

`int lastIndexOf(String <строка1>, int <начальный_индекс>)` – возвращает позицию (индекс) последнего вхождения `строки1` в данную строку. Поиск вхождения осуществляется с указанного начального индекса. Если не найдено – возвращает –1.

`boolean startsWith(String <строка1>)` – определяет, начинается ли данная строка с подстроки `строка1`.

`boolean endsWith(String <строка1>)` – определяет, заканчивается ли данная строка подстрокой `строка1`.

`String replace(char <старый_символ>, char <новый_символ>)` – заменяет в данной строке все вхождения старого символа на новый символ и возвращает полученную строку (`String`).

`String substring(int <начальный_индекс>, int <конечный_индекс>)` – возвращает часть строки, начиная с указанного начального индекса до конечного индекса.

`String trim()` – возвращает данную строку без пробелов в начале и конце строки.

`String toUpperCase()` – переводит все символы строки в верхний регистр и возвращает полученную строку.

`String toLowerCase()` – переводит все символы строки в нижний регистр и возвращает полученную строку

Примечание: обратите внимание, что операции обработки строки **не меняют исходную строку**, а лишь используют ее в качестве параметра и возвращают полученный результат.

Варианты заданий

Общие требования:

- Исходная строка и другие исходные данные (если требуется по заданию) задаются в программе при объявлении переменных или пользователем путем ввода с клавиатуры.
- Вывести на экран исходную строку и результат
- Во второй части задания «словом» считаем часть строки, ограниченную пробелами.

ВАРИАНТ 1

а) Выяснить, имеются ли пары соседствующих одинаковых символов и оставить только по одному из пары.

б) Из двух предложений удалить слова, встречающиеся в обоих предложениях. Вывести на экран полученные предложения и удаленные слова.

ВАРИАНТ 2

а) Преобразовать строку символов, заменив в ней каждую из групп, стоящих рядом точек, одной точкой.

б) Расположить слова данного предложения в порядке убывания длин.

ВАРИАНТ 3

а) Выяснить, имеется ли в строке определенная комбинация символов: s[i], s[i+1], и, если есть, сколько раз встречается.

б) Записать слова предложения в обратном порядке, исключив запятые «,»

ВАРИАНТ 4

а) Получить номера всех вхождений в строку пар одинаковых символов.

б) Найти слова, имеющие наибольшее число вхождений в предложении.

ВАРИАНТ 5

а) Ввести строку, в которой по крайней мере один восклицательный знак (не первый!). Среди символов, предшествующих "!", определить количество пробелов;

б) Даны два предложения, причем второе состоит из слов первого, записанных в другом порядке. Найти этот порядок.

ВАРИАНТ 6

а) Удалить из заданной строки все вхождения другой заданной строки.

б) Из заданного текста выбрать и напечатать те слова, которые встречаются в нем ровно один раз.

ВАРИАНТ 7

а) Преобразовать заданную строку, удалив каждый символ "*" и повторив каждый символ, равный "!".

б) В заданном предложении имеются два слова, одно из которых является обращением другого (т.е. "перевертышем" -> "нос" - "сон"). Найти эту пару слов.

ВАРИАНТ 8

а) Задать строку, в которой есть два двоеточия. Получить все символы, расположенные между первым и вторым двоеточиями.

б) Найти множество всех слов, которые встречаются в каждом из двух заданных предложений.

ВАРИАНТ 9

а) В заданной строке символов выделить в отдельную строку все имеющиеся цифры;

б) Для каждого слова заданного текста указать, сколько раз оно встречается в тексте. Сообщение об одном слове должно печататься не более одного раза.

ВАРИАНТ 10

а) В заданной строке символов перевернуть (прочитать наоборот - справа налево) определенную по двум индексам группу символов.

б) В предложении поменять местами слова с максимальной и минимальной длинами.
