Строки представляют собой набор символов и знаков. В языке программирования Java строка имеет ссылочный тип и относится к не примитивным типам данных. Для объявления строки используется зарезервированное слово String, а строка заключена в двойные кавычки. Например, строка “Hello, World” имеет 11 символов и один пробел. Объявление строки будет выглядеть так:

String anyString = "Hello, World";

Строковый тип имеет несколько особенностей:

· изменить символ в строке невозможно;

· обращаться к символам можно как к элементу массива по индексу, первый символ имеет индекс 0;

Строка может представлять собой длинную последовательность символов (текст). Строка может содержать один или ноль символов.

String anyString = "Hello, World Hello, World Hello, World Hello, World ";
String anyString = "";
String anyString = "H";

Строка может быть **null**. Это означает, что значение не было присвоено.

String anyString = null;

Библиотека Java имеет множество методов для работы со строками, вот лишь часть:

- isEmpty() возвращает **true**, если строка пуста, иначе – **false**;

- toUpperCase() возвращает новую строку в верхнем регистре;

- toLowerCase() возвращает новую строку в нижнем регистре;

- startsWith(prefix) возвращает **true**, если строка начинается с заданного префикса строки, в противном случае **false**;

- endsWith(suffix) возвращает **true**, если строка заканчивается заданным строковым суффиксом, в противном случае **false**.

- contains(...) возвращает **true**, если строка содержит заданную строку или символ;

- substring(beginIndex, endIndex) возвращает подстроку строки в диапазоне: **beginIndex**, **endIndex - 1**;

- replace(old, new) возвращает новую строку, полученную путем замены всех вхождений **old** на **new**, которые могут быть символами или строками.

- trim() возвращает копию строки, полученной путем исключения начального и конечного пробелов. Обратите внимание, что пробел включает в себя не только символ пробела, но в основном все, что выглядит пустым: табуляция, возврат каретки, символ новой строки и т. д.

Две строки могут быть объединены с помощью оператора «+» или concat метода. Оба подхода приводят к одним и тем же результатам:

```
String firstString = "Hello";
String lastString = "World";
String finalString1 = firstString + " " + lastString;
String finalString2 = firstString.concat(" ").concat(lastString);
System.out.println(finalString1);
System.out.println(finalString2);
```

Вывод на консоль будет следующий:

Hello, World

Задачи

1. Чему будет равно значение **length** после выполнения данного кода:

```
String str = " Hello World";

String subString = str.substring(0, 7).trim();

int length = subString.length();
```

Ответ 5, первый метод **substring** вернет подстроку длиной 7 символов **" Hello ",** а метод **trim()** удалит пробелы в начале и в конце строки, таким образом уменьшив строку на два символа.

2. Чему будет равно значение **lastString** в приведенном ниже коде:

```
String firstString = "Hello";

String lastString = firstString;

firstString = "By";
```

Ответ "Hello", строка это ссылочный тип, переменная **lastString** ссылается на строку **"Hello",** изменение ссылочной переменной **firstString** не изменяет **lastString.**

3. Выберите все верные утверждения о строках в Java:

   a) Строка — это примитивный тип данных.

   b) Строка имеет метод с именем **size()**, который возвращает количество символов в этой строке.

   c) Две строки могут быть объединены с помощью операции **+**.

   d) Строка — это последовательность символов.

e) Строки неизменяемы.

Ответ c, d, e.

4. Выберите правильный вариант создания строки:

    a) String s = "Wor" + "l" + 'd';

    b) String s = "World";

    c) String s = 'World';

Ответ b. Строки заключены в двойные кавычки.

5. Выберите правильные действия над строками:

    a) s1 + s2;

    b) s1 - s2;

    c) s1 * s2;

    d) s1 < s2;

Ответ d. Для объединения строк используется оператор +.