

String. Ввод/Вывод

String

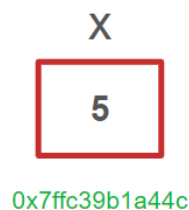
String в Java - это ссылочный (объектный) тип данных, предназначенный для представления строк. По структуре, объект String представляет собой упорядоченный набор символов (массив) типа `char`. Отсчет символов в строке начинается с нуля, то есть первый символ имеет индекс 0.

```
String s = "Hello";
```

| | | | | |
|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 |
| н | е | л | л | о |

Тип данных String является ссылочным. Следовательно, переменная типа String называется объектной переменной и содержит адрес строки в памяти, а не саму строку. В этом главное отличие между примитивными и ссылочными типами.

```
int x = 5;
```



```
String s = "Hello";
```



Строковые литералы в Java должны быть заключены в двойные кавычки, например: `"Hello, World!"`. Это отличает строковые литералы от литералов других типов, например, символьных литералов (`char`), которые заключаются в одинарные кавычки.

В Java можно **соединять** (конкатенировать) строки с помощью оператора `+`. При конкатенации строки с переменной другого типа (например, `int`, `double`, `boolean`), последняя автоматически преобразуется в строку.

```
int x = 5;
double y = 2.5;
char c = 'A'

String s = "Hello";
String s1 = s + x; // Hello5
String s2 = s + y; // Hello2.5
String s3 = s + c; // HelloA
```

Метод **`charAt(int index)`** позволяет получить символ, расположенный по указанному индексу в строке:

```
String sample = "Hello";
char letter = sample.charAt(1); // letter будет 'e'
```

Метод **`indexOf(String str)`** позволяет получить индекс первого вхождения указанной подстроки в строке. Если подстрока не найдена, результатом будет **-1**.

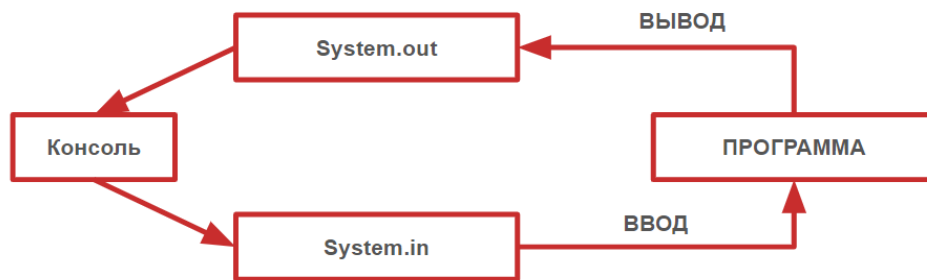
```
String sample = "Hello, World!";
int index = sample.indexOf("World"); // index будет 7
```

Ввод/вывод

В Java, при выполнении приложения, по умолчанию открыты два основных потока для взаимодействия с консолью: поток ввода **`System.in`** и поток вывода **`System.out`**.

Эти потоки предоставляют базовый механизм для ввода данных из консоли и вывода данных в консоль соответственно. `System.out` является стандартным потоком вывода и обычно используется для отображения результатов программы на консоли.

Когда вы используете команды, такие как `System.out.println("text")`, вы фактически отправляете строку `"text"` в поток `System.out`, который затем выводит её в консоль. Этот поток особенно полезен для отладки, так как позволяет разработчикам отслеживать ход выполнения программы и выводить различные сообщения и значения переменных.

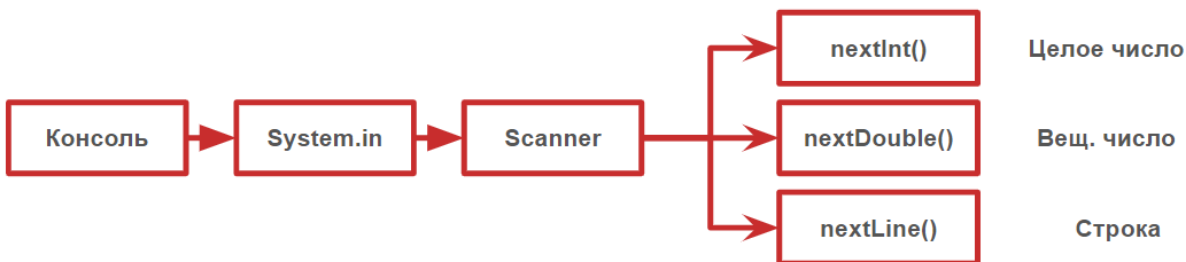


С другой стороны, `System.in` является стандартным потоком ввода в Java, который обычно используется для чтения данных, вводимых пользователем через консоль. Данные, введенные в консоль, могут быть прочитаны и обработаны программой через `System.in`.

Например, можно использовать класс `Scanner`, который обортывает `System.in`, для удобства чтения вводимых данных, таких как целые числа, строки, и другие типы данных.

Таким образом, `System.out` и `System.in` играют ключевую роль в создании интерактивных консольных приложений в Java, позволяя программам общаться с пользователем и обрабатывать его ввод, а также выводить результаты работы или сообщения об ошибках. Эти потоки обеспечивают основной интерфейс для взаимодействия между приложением и внешним миром через консоль.

Scanner



Scanner в Java - это класс, предоставляющий удобные методы для чтения входных данных, в том числе из стандартного ввода, представленного `System.in`. Этот класс используется для разбора примитивных типов данных и строк из потоков ввода. При создании экземпляра `Scanner`, он может быть инициализирован `System.in`, что позволяет `Scanner` читать вводимые данные.

- **nextLine():** Читает строку ввода. Он читает весь ввод, включая пробелы, до тех пор, пока не встретит символ новой строки (нажатие Enter).
- **nextInt():** Читает следующий токен ввода как `int`.
- **nextDouble():** Читает следующий токен ввода как `double`.

// Чтение строки

```
System.out.print("Введите строку: ");  
String line = scanner.nextLine();
```

// Чтение целого числа

```
System.out.print("Введите целое число: ");  
int number = scanner.nextInt();
```

// Чтение вещественного числа

```
System.out.print("Введите вещественное число: ");  
double dNumber = scanner.nextDouble();
```