

Профессия Java-разработчик за 12 месяцев

Записывайся на новый курс от JavaRush!



Обучение
в группе



Видеоуроки с опытными
преподавателями



Практические
домашние задания



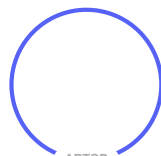
Дополнительные
задачи и проекты

Поспеши: в группе осталось несколько мест.

ПОДРОБНЕЕ

Статьи Авторы Все группы Все статьи

[JavaRush](#) / [Java блог](#) / [Random](#) / Метод IndexOf класса String: зачем он нужен и как работае...



АВТОР

Василий Малик

Senior Java-разработчик в
CodeGym



19 августа 2020



55236



19

Метод IndexOf класса String: зачем он нужен и как работает

Статья из группы Random

Присоединиться

Привет!

Программисты часто пользуются методами на автомате, не задумываясь, как они работают. Речь идет о базовых методах, которые вы обязаны знать.

Что такое перегрузка методов?

Прежде, чем мы рассмотрим метод, о котором упомянули в заголовке, давайте вспомним (а кто-то и узнает), что такое перегрузка методов.

Перегрузка методов Java разрешает определение внутри одного класса двух или более методов с одним и тем же именем. Это возможно только в том случае, если у них есть разные входящие параметры. В этом случае методы и называют перегруженными, а сам процесс — перегрузкой методов.





Перегрузка методов — это один из основополагающих принципов [полиморфизма](#) в Java.

Для чего нужен метод `indexOf` класса `String`

Представьте, что у вас есть длинная строка. К примеру, письмо, и вам нужно найти место, в котором происходит первое обращение к адресату по имени, которое вы уже знаете. Как вы это сделаете? Воспользуетесь методом `indexOf` java класса `String`.

Этот метод — типичный пример перегрузки методов, о которой мы говорили выше.

Варианты метода `indexOf`

Метод `java string indexOf` имеет четыре различные вариации:

→

**Получите профессию
Java-разработчика**

на онлайн-курсе с ментором
и сертификацией

Онлайн-лекции с опытными менторами

Групповое обучение и поддержка в закрытом чате

10 проектов в вашем портфолио и сотни часов кодига

Помощь в поиске первой работы

Скоро стартуют занятия в новой группе – поспешите!

Подробнее ↗

№ 1. `indexOf(int ch)`

Метод `int indexOf(int ch)` возвращает индекс в данной строке первого вхождения указанного символа. Другими словами, мы получим номер первого вхождения заданного символа, считая слева-направо.

Пример:

```
1 public static void main(String[] args) {
2     String str = "Diego, where is my money?";
3     int value = str.indexOf('e');
4     System.out.println(value);
5 }
```

Вывод в консоль:

2

Если же символ который мы ищем, отсутствует в данной строке, мы получим -1.

```
1 public static void main(String[] args) {  
2     String str = "Diego, where is my money?";  
3     int value = str.indexOf('j');  
4     System.out.println(value);  
5 }
```

Вывод в консоль:

-1

P. S. Чтобы ничего не перепутать, нужно не забывать, что отсчет символов в строке идет не с «1,2,3...», а с «0,1,2...»

№ 2. `indexOf(int ch, int fromIndex)`

Метод `int indexOf(int ch, int fromIndex)` возвращает индекс в этой строке первого вхождения указанного символа, начиная поиск по указанному индексу.

Данный метод является более модернизированной версией предыдущего. С разницей в том, что мы указываем номер элемента, с которого собственно и начнется поиск.

Пример:

```
1 public static void main(String[] args) {  
2     String str = "Diego, where is my money?";  
3     int value = str.indexOf('e', 5);  
4     System.out.println(value);  
5 }
```

Вывод в консоль:

9

Если индекс, из которого будет вестись поиск, превышает длину строки, то ответ будет -1.

