

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

Поспеси: в группе осталось несколько мест.

ПОДРОБНЕЕ

Карта квестов

Лекции

Открытые квесты

JavaRush / Java Курсы / Java Multithreading / Внутреннее устройство String, метод substring

Внутреннее устройство String, метод substring

Java Multithreading
2 уровень , 3 лекция

ОТКРЫТА



— Привет, Амиго!

— Привет, Элли!

— В этом уроке я тебе расскажу о двух вещах. А конкретно — о подстроках и об устройстве объектов класса String. Причём в этом вопросе нам придётся немного погрузиться в историю Java... Но это будет коротко, интересно и полезно, обещаю.

Как получить подстроку (или часть строки)?

Из поставленного вопроса ты уже, вероятно, понял, что подстрока — это просто часть строки. Знаешь, какие действия над строками самые популярные? Первое — склеивание нескольких строк (конкатенация), ты уже с ней сталкивался неоднократно. И второе — получение подстроки из строки.



В Java есть метод **substring**. Он возвращает часть строки. Существует два варианта этого метода.

Первый возвращает подстроку, заданную начальным и конечным номерами символов. Обрати внимание, **первый символ при этом входит в подстроку, а последний — нет!** То есть, если мы передаем номера (1,3) — с первого по третий, то в подстроке окажутся только первый и второй символы.

Во втором варианте методы мы указываем только начальный индекс подстроки, и он возвращает подстроку от этого индекса до конца строки.

Метод(ы)	Пример(ы)
<div><div>1</div><div>String substring(int beginIndex, int endIndex)</div></div>	<div><div><div>1</div><div>String s = "Good news everyone!";</div></div><div><div>2</div><div>s = s.substring(1,6);</div></div></div> <div>Результат:</div> <div><div><div>1</div><div>s == "ood n";</div></div></div>
<div><div>1</div><div>String substring(int beginIndex)</div></div>	<div><div><div>1</div><div>String s = "Good news everyone!";</div></div><div><div>2</div><div>s = s.substring(1);</div></div></div> <div>Результат:</div> <div><div><div>1</div><div>s == "ood news everyone!";</div></div></div>

— Все достаточно просто. Спасибо, Элли.

Внутреннее устройство объектов String: экскурс в JDK 6

— Помнишь, я тебе обещала небольшой экскурс в историю Java? Разумеется, в контексте нашей темы, а именно — особенностей класса String.

Когда-то давно, в те времена тебя, возможно, ещё и не собрали, самой актуальной версией языка была JDK 6. С тех пор много воды утекло и число, обозначающее номер самой новой Java уже давно стало двузначным.

Как ты уже знаешь из первой лекции 2-го уровня этого квеста, String — неизменяемый класс (immutable). И вот эта самая неизменяемость дала возможность получать подстроку интересным способом в JDK 6.

Внутри объект типа String является массивом символов, а точнее — содержит массив символов. Это интуитивно понятно. А во времена JDK 6 там хранились еще две переменные: номер первого символа в символьном массиве и их количество. Таким образом, в JDK 6 у String было три поля **char value[]** (символьный массив), **int offset** (индекс первого символа в массиве) и **int count** (количество символов в массиве).

Это устройство и было использовано для создания подстроки с помощью метода **substring()**. Когда он вызван, он создает новую строку, то есть новый объект String.

Только вот вместо того, чтобы хранить ссылку на массив с новым набором символов, в JDK 6 этот объект хранил ссылку на старый массив символов и ещё две переменные **offset** и **count**. С их помощью он определяет, какая часть оригинального массива символов относится к новому подмассиву.

— Ничего не понял.