

آموزش زبان برنامه نویسی جاوا

نمونه مثال از کلاس String

متد compareTo()

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



متد `compareTo()` دو `String` (رشته) را از نظر لغوی مقایسه می کند. ابتدا کاراکترهای (`character`) هر دو `String` (رشته) به مقدار `Unicode` (یونیکد) برای مقایسه با هم تبدیل می شوند.

اگر دو `String` (رشته) باهم برابر بودند این متد مقدار `0` (صفر) را برمیگرداند، اگر `String` (رشته) اول از نظر لغوی بزرگتر از `String` (رشته) دوم بود یک مقدار مثبت را برمیگرداند و در غیر این صورت یعنی اگر `String` (رشته) اول از نظر لغوی کوچک تر از `String` (رشته) دوم بود، یک مقدار منفی را برمی گرداند.

## متد `compareTo` با داشتن پارامتری از نوع `String literal`:

```
int compareTo(String str)
```

این متد برای مقایسه دو `String` (رشته) که به روش `String literal` ایجاد شده اند، استفاده می شود.

نکته مهم: پارامتر این متد برای مقایسه دو `String literal` می باشد.

میدونم احتمالا `String literal` به گوشتون نخورده باشه!

## String literal

در کل ما دو روش برای ایجاد `String` (رشته) در جاوا در اختیار داریم:

۱. `String literal` (رشته لیترال ها)

۲. `Using new keyword` ( ساختن شی از کلاس `String` از طریق کلمه کلیدی `new` )

نکته: اصلاحات انگلیسی رو یادبگیرید بهتر از فارسی هست چون میتونید برای درک بهتر مفاهیم در گوگل سرچ بزنید.

## • String literal چیست؟

به زبان ساده براتون می‌گم! اگر در جاوا Strings (رشته‌ها) را به صورت زیر تعریف کردید ، به اون Strings (رشته‌های) ایجاد شده String literal می‌گوییم:

```
String str1 = "Welcome";  
String str2 = "Welcome";
```

- به زبان ساده وقتی یک String را شبیه به روش متغیرهای معمولی در جاوا تعریف می‌کنیم بهش String literal می‌گوییم.

همان طور که در جلسه آموزشی کلاس String بررسی کردیم و می‌دانید که String یک کلاس است و برای استفاده از آن باید شی ایجاد کنیم. اما ما در روش بالا هیچ شی از کلاس String ایجاد نکردیم! پس چرا برنامه خطا نمیده؟! چون compiler (کامپایلر) وظیفه ایجاد شی از کلاس String را در روش بالا برای ما انجام می‌دهد. یعنی از یک String literal یک شی از نوع کلاس String ایجاد می‌کند.

اما اگر شی از قبل در حافظه وجود داشته باشد، کامپایلر یک شی جدید ایجاد نمی‌کند و تنها یک شی قدیمی را به نمونه جدید می‌دهد، مثلا برای دو نمونه String (رشته) بالا یعنی str1 و str2 که مقدار یکسانی دارند کامپایلر تنها یک شی (object) با مقدار "welcome" ایجاد می‌کند و برای هردو اختصاص می‌دهد. لازم است بدانید که کامپایلر تنها برای Strings (رشته‌های) با مقدار یکسان تنها یک شی در روش String literal ایجاد می‌کند.

برای مثال فرض کنید ۱۰ نمونه String (رشته) با مقدار یکسان داشته باشیم، بدین ترتیب در حافظه فقط یک شی دارای همان مقدار یکسان برای هر ۱۰ نمونه String اختصاص داده می‌شود. به عبارتی ما ۱۰ شی مجزا از نوع String نخواهیم داشت. چرا؟ چون ما ۱۰ نمونه String با مقدار یکسان را از روش String literal ایجاد کرده ایم.

اگر بخواهیم ۱۰ شی مجزا با مقدار یکسان ایجاد کنیم چاره چیست؟!

پاسخ: برای این کار باید String های خود را با استفاده از کلمه کلیدی new ایجاد کنید. در جلوتر این روش را بررسی می‌کنیم.

برگردیم به متد `int compareTo(String str)` ، در زیر مثالی از دو `String` که به روش `String literal` ایجاد شده اند و قراره با هم مقایسه شوند، آورده ایم:  
فرض کنید دو `String` (رشته) بصورت زیر داشته باشیم:

```
String string1="java tutorial";  
String string2="persian java tutorial";
```

برای مقایسه این دو `String` (رشته) با استفاده از متد `int compareTo(String str)` به صورت زیر عمل می کنیم:

```
string1.compareTo(string2);
```

#### • Using new keyword (ساختن شی از کلاس `String` از طریق کلمه کلیدی `new`)

اگر در جاوا `Strings` (رشته ها) را به صورت زیر تعریف کردید ، به اون `Strings` (رشته های) ایجاد شده شی از کلاس `String` می گوییم:

```
String str1 = new String("Welcome");  
String str2 = new String("Welcome");
```

متد `compareTo` با داشتن پارامتری از نوع کلاس `Object` :

```
int compareTo(Object obj)
```

برای مقایسه یک شی از نوع کلاس `String` با یک `String literal` از این متد با این نوع پارامتر استفاده می کنیم.

فرض کنید یک `String literal` به صورت زیر داشته باشیم:

```
String str1 = "Welcome";
```

و یک شی از نوع کلاس String به صورت زیر ایجاد کرده باشیم:

```
String str2 = new String("Welcome");
```

برای مقایسه str1 با str2 از متد (int compareTo(Object obj)) استفاده می کنیم:

```
str1.compareTo(str2)
```

مثال:

در مثال زیر ما سه String داریم که با استفاده از متد compareTo آنها را مقایسه کرده ایم.

```
package javalike;

public class CompareToExample {
    public static void main(String args[]) {
        String str1 = "String method tutorial";
        String str2 = "compareTo method example";
        String str3 = "String method tutorial";

        int var1 = str1.compareTo(str2);
        System.out.println("str1 & str2 comparison: " + var1);

        int var2 = str1.compareTo(str3);
        System.out.println("str1 & str3 comparison: " + var2);

        int var3 = str2.compareTo("compareTo method example");
        System.out.println("str2 & string argument comparison: " + var3);
    }
}
```

خروجی (Output):

```
str1 & str2 comparison: -16
str1 & str3 comparison: 0
str2 & string argument comparison: 0
```

پیروز و موفق باشید

این جلسه آموزشی رایگان است، فروش و ویرایش آن ممنوع و حرام می باشد. اما این کتاب را می توانید همین جور که هست در سایت و شبکه اجتماعی خود به اشتراک بگذارید.

[www.JAVAPRO.ir](http://www.JAVAPRO.ir)

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

## بازدید از کانال

## بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.