

آموزش زبان برنامه نویسی جاوا

(کار با فایل)

کلاس `BufferedReader`

جلسه سی و سوم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



- راه های زیادی برای خواندن داده ها از فایل وجود دارد که در جلسات گذشته بخشی از آنها را بررسی کردیم. در این جلسه قصد داریم ساده ترین راه خواندن اطلاعات را از فایل بررسی کنیم.
- راحت ترین و متداول ترین راه خوانده داده ها از یک فایل استفاده از کلاس **BufferedReader** می باشد.
- کلاس **BufferedReader** بسیار کارآمد برای خواندن کاراکترها، آرایه ها و سطر های درون یک فایل می باشد.

## BufferedReader

برای استفاده از این کلاس در برنامه من باید پکیج **java.io.BufferedReader** را import کنیم.

```
import java.io.BufferedReader;
```

یکی از متداول ترین سازنده های این کلاس که باهش کار داریم بصورت زیر است:

### BufferedReader(Reader in)

این سازنده هنگام شی سازی از این کلاس بعد از کلمه کلیدی **new** صدا زده می شود:

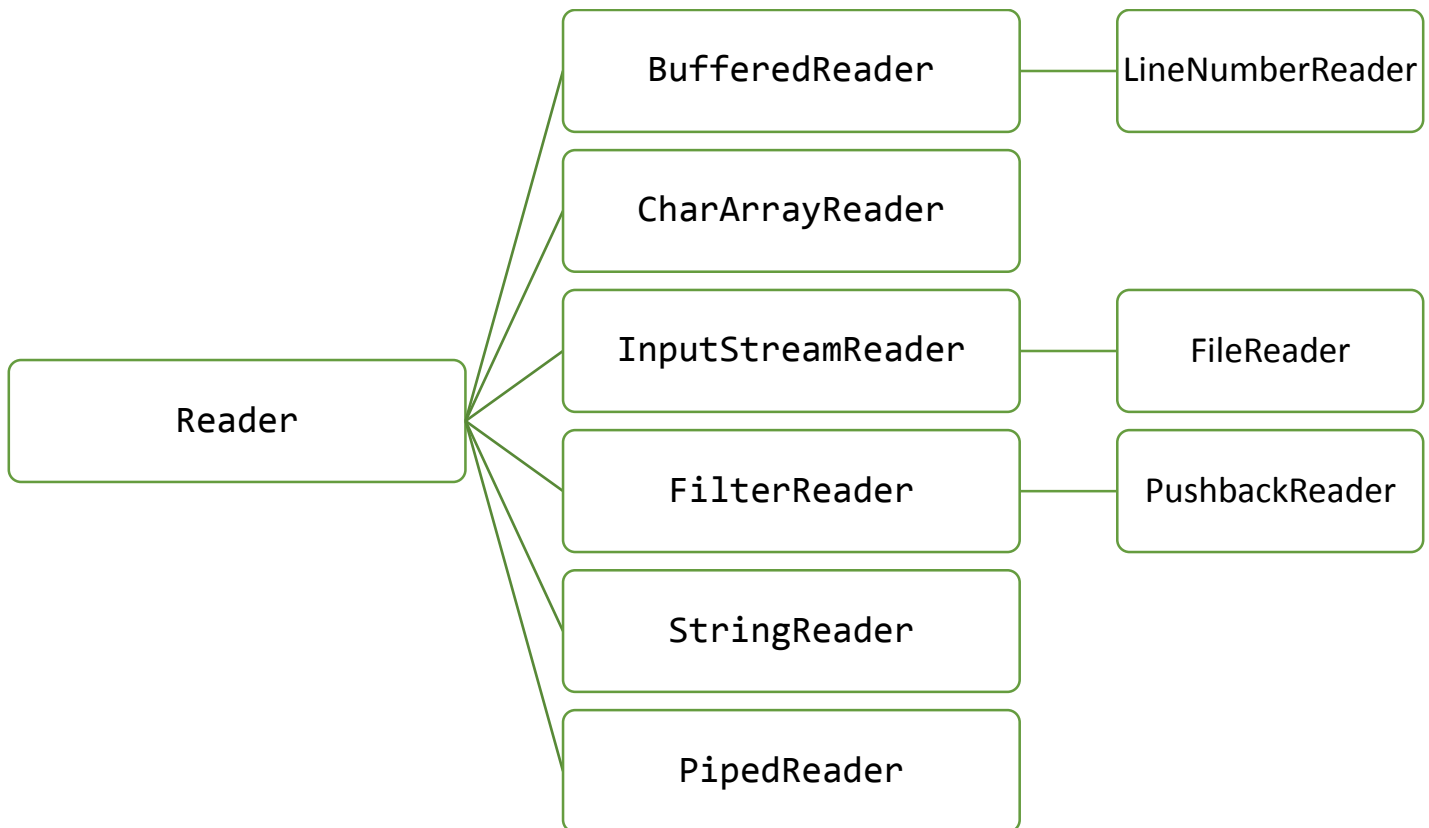
```
BufferedReader b = new BufferedReader(Reader in)
```

پارامتر این سازنده که در اینجا نامش **in** هستش از نوع کلاس **Reader** تعریف شده است.

خب اولین سوال که برامون پیش میاد کلاس **Reader** چیه؟!؟

کلاس **Reader** یک کلاس از نوع انتزاعی (**abstract**) هستش که پدر برخی از کلاس هایی است که در زمینه خواندن فایل در جاوا استفاده می کنیم!!

سلسه مراتب کلاس **Reader** را در زیر مشاهده می کنید: (تصویر ۱)



تصویر (۱)

تمام اشیا ایجاد شده از کلاس هایی که فرزند (زیر کلاس) کلاس Reader در سلسله مراتب بالا هستند، می توانند به عنوان پارامتر بجای **Reader in** در سازنده کلاس **BufferedReader** قرار بگیرند.

مثال:

```
FileReader fr=new FileReader("test.txt");
```

```
BufferedReader b = new BufferedReader(fr);
```

همون طور که می بینید ما بجای پارامتر موجود در سازنده کلاس **BufferedReader** یک شی از نوع **FileReader** قرار داده ایم که فرزند کلاس **Reader** می باشد.

خب این چه کاری هست!! چرا از خود کلاس **Reader** شی نمی سازیم؟! پاسخ: دلیل این که از کلاس **Reader** استفاده نمی کنیم این است که کلاس **Reader** انتزاعی و **Abstract** هستش و از کلاس **Abstract** نمی توان شی ساخت!!!

**نتیجه گیری کلی:** ما به جای پارامتر موجود در سازنده کلاس **BufferedReader** می توانیم از اشیا تمام کلاس هایی که فرزند کلاس **Reader** هستش استفاده کنیم.

## پدرسی متدهای کلاس `BufferedReader` :

### `void close()`

این متد برای بستن فایل است که با استفاده از کلاس `BufferedReader` آن را خوانده ایم. به یاد داشته باشید بعد از این که کارمون با فایل خوانده شده تمام شد با این متد آن را ببندیم.

### `int read()`

یک متد نام آشنا!!!! این متد بصورت کاراکتر به کاراکتر فایل ما رو میخواند. کاراکترهای خوانده شده را بصورت معادل عددی هر کاراکتر از نوع عدد صحیح بر می گرداند. مثلا اگر در فایل ما حرف `a` بود معادل عددی این حرف یعنی عدد `۹۷` را برمی گرداند.

خب تا اینجا متدهای این کلاس فرقی با سایر کلاس هایی که در زمینه خواندن فایل در جاوا بود نداشت!!!!!!!!!!!!!! صبر کنید متد زیر رو ببینید!!!!

### `String readLine()`

- این متد یک سطر از متن درون فایل ما را می خواند.

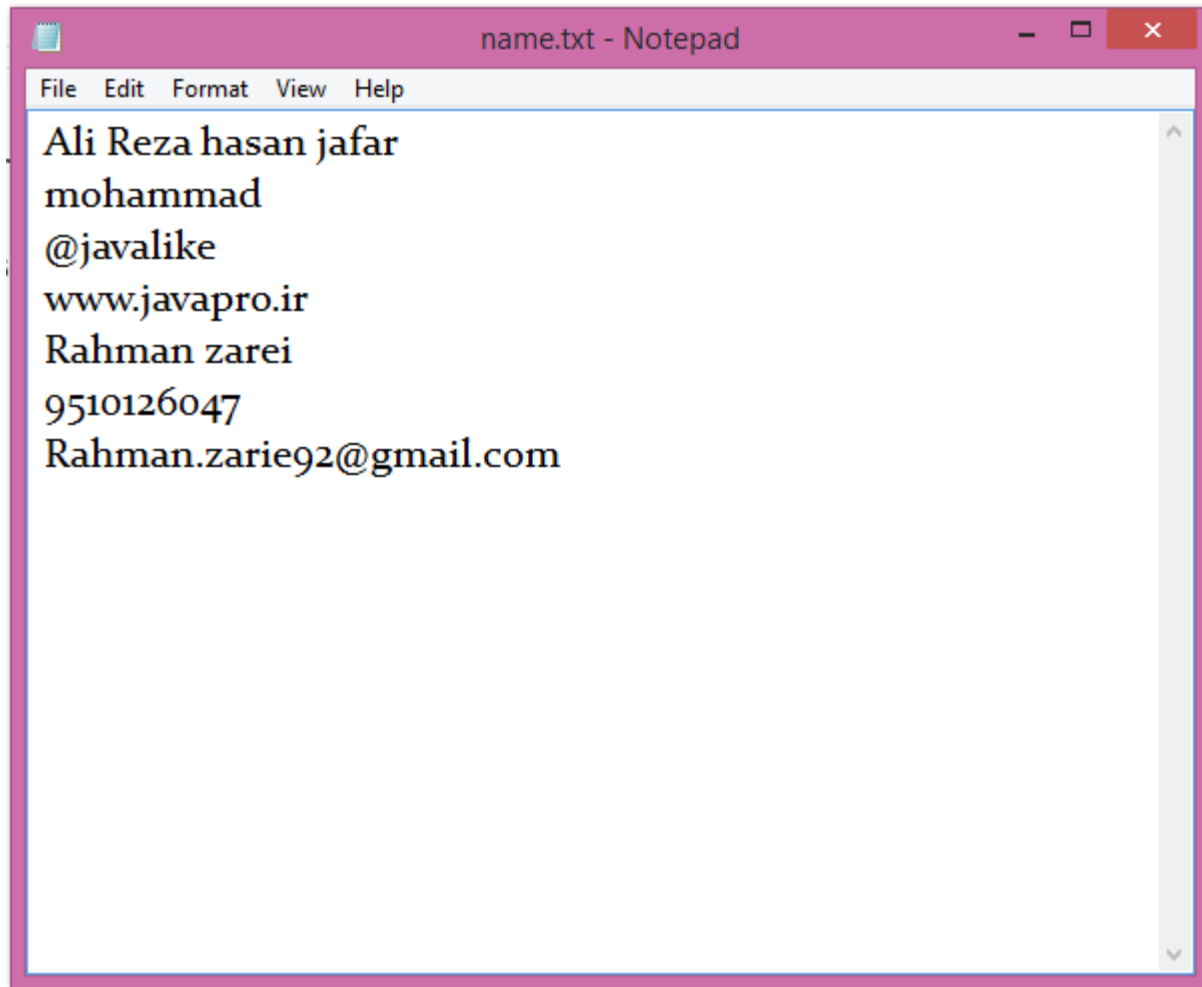
- پرکاربردترین متد برای خواندن داده های درون یک فایل می باشد.

- متد `readLine` کار ما را برای خواندن داده های درون فایل ساده کرده است. چون مقداری که برمیگرداند یک سطر

از متن درون فایل بصورت رشته است و دیگه نیاز نیست کاراکتر به کاراکتر فایلمون رو بخوانیم. در مثال کاربرد این

متد را بررسی می کنیم.

مثال: فرض کنید در کامپیوتر، پوشه پروژمون فایل با نام و فرمت `name.txt` داریم که متن درون آن بصورت تصویر (۲) است:



تصویر (۲)

متن فایل در تصویر (۲) را مشاهده کنید، محتوای درون این فایل یک متن هفت سطری می باشد، ما قصد داریم با استفاده از کلاس `BufferedReader` اطلاعات درون این فایل را بخوانیم:

```
package www.javapro.ir;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;

import fileIO.fileStreamTest;
```

```
public class BufferedReaderDemo {
    public static void main(String[] args) throws Exception {

        try {
            // open input stream name.txt for reading purpose.

            FileReader in = new FileReader("name.txt");
            BufferedReader br = new BufferedReader(in);

            System.out.println(br.readLine());
            Br.close();

        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

خروجی:

### Ali Reza hasan jafar

با وجودی که فایل ما از ۷ سطر تشکیل شده بود تنها یک سطر از فایل ما خوانده شده و در محیط کنسول چاپ شده است!!!!  
دلیلش اینه که با هر بار صدا زدن متد `readLine()` تنها یک سطر از فایل خوانده می شود و ما در مثال بالا فقط یکبار این متد را صدا زده ایم و در نتیجه تنها یک سطر از فایل ما خوانده شده است.  
برای خواندن تمامی سطرهای فایلمون بصورت زیر عمل می کنیم:

```
package www.javapro.ir;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileReader;
import java.io.InputStream;
import java.io.InputStreamReader;

import fileIO.fileStreamTest;

public class BufferedReaderDemo {
    public static void main(String[] args) throws Exception {

        try {
            // open input stream name.txt for reading purpose.
```

```
String thisLine = new String();

FileReader in = new FileReader("name.txt");
BufferedReader br = new BufferedReader(in);

while ((thisLine = br.readLine()) != null) {

    System.out.println(thisLine);
}

    Br.close();

} catch (Exception e) {
    e.printStackTrace();
}
}
```

خروجی:

Ali Reza hasan jafar  
mohammad  
@javalikey  
www.javapro.ir  
Rahman zarey  
9510126047  
Rahman.zarie92@gmail.com

- در این برنامه با قرار دادن یک حلقه سطر به سطر اطلاعات درون فایلمون رو خونديم.

```
String thisLine = new String();
```

- تعريف يك متغير از نوع String ، هر سطر از فايل بعد از خوانده شدن در اين متغير ريخته مي شود.

```
FileReader in = new FileReader("name.txt");
BufferedReader br = new BufferedReader(in);
```

- براي خواندن فايل از كلاس BufferedReader استفاده شده است. همان طور كه گفتيم پارامتر سازنده اين كلاس بايد از نوع يكي از كلاس هاي فرزند Reader باشد كه ما در اينجا كلاس FileReader را انتخاب كرديم و آدرس محل فايل در كامپيوترمون رو بهش داده ايم.
- وقتي آدرس محل فايل بصورت يك اسم همراه با يك فرمت مثل آدرس زير باشد يعني محل فايل ما درون پوشه پروژه سورش كد ما مي باشد:

```
"name.txt"
```

```
while ((thisLine = br.readLine()) != null) {
```

```
System.out.println(thisLine);  
}
```

این حلقه سطر به سطر متن درون فایل ما را می خواند، خب الگوریتم و روش کارش چطوره؟!  
پاسخ :

می دونید که با هر بار صدا زدن متد `readLine()` یک سطر از متن درون فایل خوانده می شود.  
ما از یک حلقه برای تعداد تکرار صدا زدن این متد استفاده کردیم به این صورت که تا هنگامی که متد `readLine()` به یک سطر `null` نرسیده کار خواندن سطر به سطر فایل را ادامه دهد و این متد یعنی `readLine()` هر بار که یک سطر از متن درون فایل را می خواند درون متغیر `thisLine` می ریزد و بعدش در بدنه حلقه دستور چاپ مقدار متغیر `thisLine` را داریم که مقدار هر سطر بعد از خوانده شدن چاپ می شود.

● بلوک `try-catch` نیز برای کنترل استثناهای احتمالی در زمینه کار با فایل می باشد.

این مبحث از کار با فایل نیز تمام شد ، لازمه یادگیری کار با فایل دیدن مثال های گوناگون و فراوان می باشد. که دارم سعی میکنم مثال هایی در همه زمینه مباحث جاوا را به امید خدا طراحی کنم و روی سایت قرار دهم.

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

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.