

آموزش زبان برنامه نویسی جاوا

کلاس BufferedWriter

جلسه سی و چهارم

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



در این آموزش قصد داریم یکی از روش های ساده ایجاد فایل و نوشتن داده ها و اطلاعات درون فایل را بررسی کنیم یکی راه های ساده و متداول ایجاد و نوشتن فایل در جاوا استفاده از کلاس **BufferedWriter** می باشد.

با کلاس **BufferedWriter** می توان جریانی از داده های کاراکتری، رشته، آرایه و غیره را در فایل ذخیره کنیم.

برای استفاده از کلاس **BufferedWriter** در برنامه خود نیاز به **import** کردن پکیج **Java.io.BufferedWriter** داریم:

```
import java.io.BufferedWriter;
```

یکی از متداول ترین سازنده کلاس **BufferedWriter** بصورت زیر است:

```
BufferedWriter(Writer out)
```

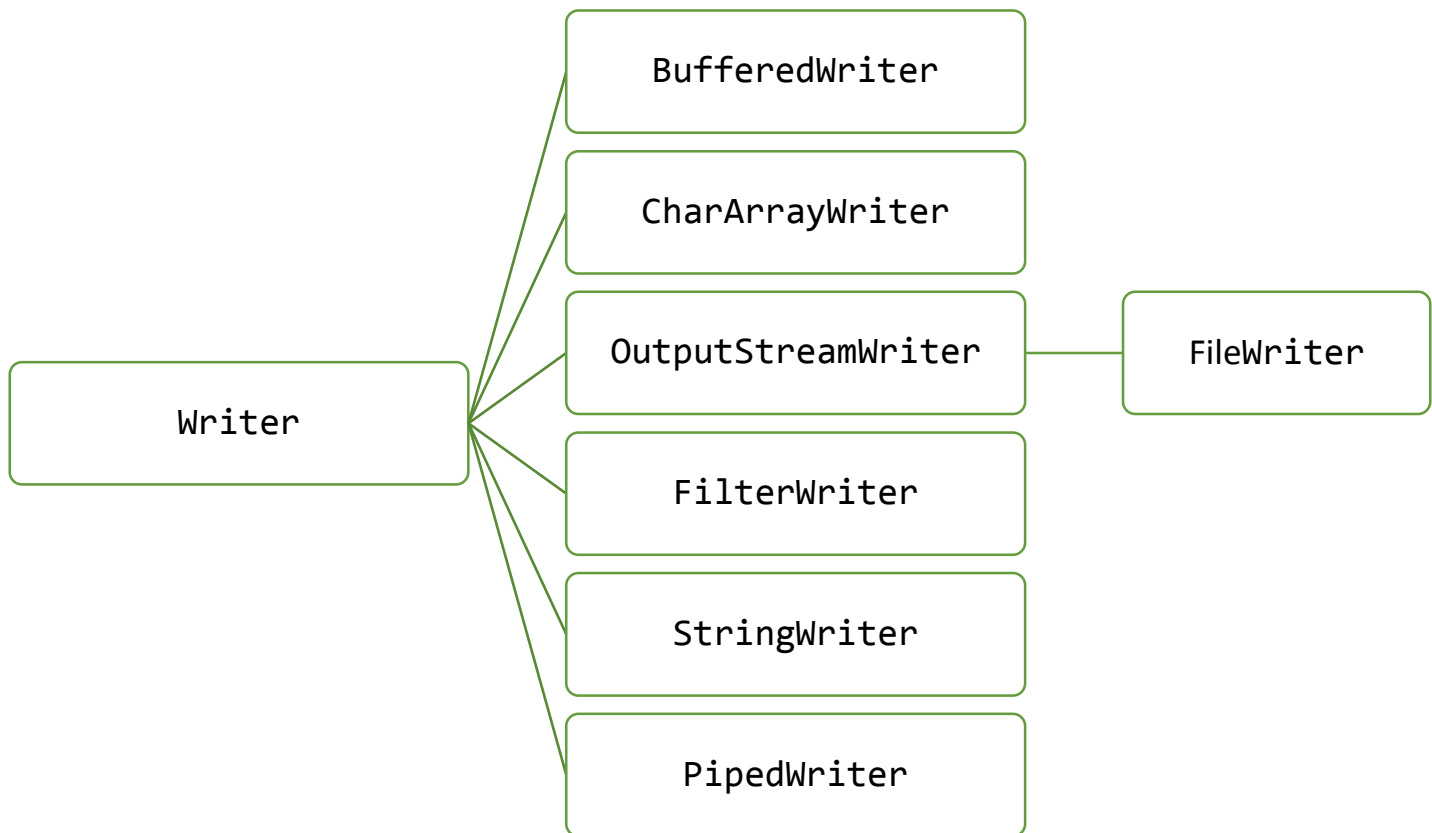
هنگام شی سازی از کلاس **BufferedWriter** بعد از کلمه کلیدی **new** این سازنده صدا زده می شود و پارامتر درون پرانتز آن مقدار دهی می کنیم.

```
BufferedWriter bw=new BufferedWriter(Writer out)
```

پارامتر سازنده کلاس **BufferedWriter** از نوع کلاس **Writer** می باشد.

میدونم الان میخوايد بگيد که من تا حالا اسم کلاس **Writer** به گوشم نخورده!! 😊

کلاس **Writer** یک کلاس از نوع انتزاعی (**abstract**) هستش که پدر برخی از کلاس هایی است که در زمینه نوشتن داده ها در فایل کاربرد دارند. در تصویر (۱) سلسله مراتب کلاس هایی که فرزند کلاس **Writer** هستند آورده شده است:



تصویر (۱)

تمام اشیایی که از نوع کلاس هایی که فرزند (زیرکلاس) کلاس **Writer** هستند و در سلسله مراتب تصویر (۱) هم نمایش داده شده است ، می توانند جایگزین پارامتر **Writer out** در سازنده کلاس **BufferedWriter** شوند.

مثلا:

```
FileWriter fw = new FileWriter("newFile.txt");  
BufferedWriter bw = new BufferedWriter(fw);
```

همان طور که در مثال بالا مشاهده می کنید یک شی از کلاس **FileWriter** که طبق سلسله مراتب تصویر (۱) از فرزندان کلاس **Writer** هستند جایگزین پارامتر موجود در سازنده کلاس **BufferedWriter** شده است.

خب حتما براتون سوال پیش اومده چرا مستقیم از کلاس **Writer** شی نمی سازیم و درون سازنده کلاس **BufferedWriter** قرار نمی دهیم؟!!!

از آنجایی که کلاس `Writer` انتزاعی (`abstract`) هستش نمی توان مستقیم از این کلاس شی ایجاد کرد، به همین دلیل از فرزندان کلاس `Writer` که به تمام ویژگی ها و رفتارهای کلاس `Writer` دسترسی دارند شی ساخته و درون سازنده کلاس `BufferedWriter` قرار می دهیم.

به زبان ساده : در کل ما هنگام شی سازی از کلاس `BufferedWriter` ، بعد از کلمه کلیدی `new` سازنده این کلاس را صدا می زنیم و به جای پارامتر سازنده این کلاس ، اشیای یکی از فرزندان کلاس `Writer` را قرار می دهیم.
نکته: فرزندان کلاس `Writer` در تصویر (۱) گردآوری شده اند.

پدرسی متدهای کلاس `BufferedReader` :

`void close()`

بعد از این که فایلمون رو ایجاد کردیم و روش عملیات های مورد نظر رو انجام دادیم و دیگه کاری باهاش نداشتیم با این متد فایلمون رو می بندیم.

`void write(String str)`

این متد برای نوشتن و ذخیره کردن داده ها درون فایل استفاده می شود.

مثال:

```
package www.javapro.ir;

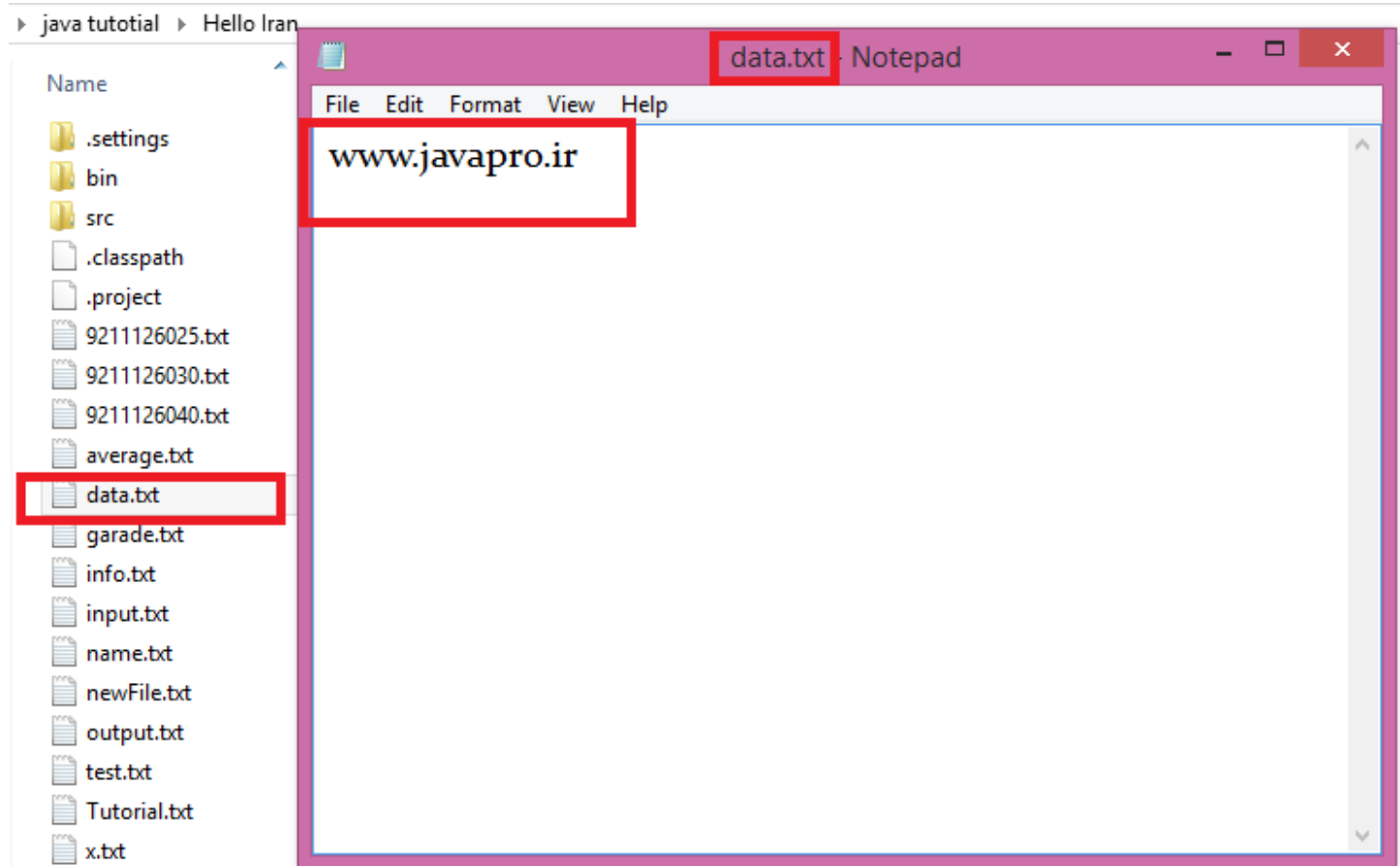
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class TestBufferedWriter {

    public static void main(String[] args) {
        try {
            BufferedWriter bw=new BufferedWriter(new FileWriter("data.txt"));
            bw.write("www.javapro.ir");
            bw.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
        }
    }
}
```

```
        e.printStackTrace();
    }
}
}
```

خروجی: در کنسول خروجی نداریم و خروجی این برنامه بصورت یک فایل با نام و فرمت `data.txt` می باشد. تصویر (۲)



تصویر (۲)

توضیحات:

```
BufferedWriter bw=new BufferedWriter(new FileWriter("data.txt"));
```

- در این خط کد ، ما برای ایجاد فایل از کلاس **BufferedWriter** استفاده کرده ایم.

- ابتدا از کلاس **BufferedWriter** شی ایجاد کردیم. درون سازنده کلاس **BufferedWriter** یک شی از نوع کلاس **FileWriter** که یکی از فرزندان کلاس **Writer** هستش شی ساختیم. بعد درون سازنده کلاس **FileWriter** آدرس مکانی که قراره فایل در آن ایجاد شود قرار دادیم

به طور کلی بصورت زیر عمل کردیم:

۱. ایجاد یک شی از نوع کلاس **BufferedWriter**
۲. بعد از کلمه کلیدی **new** سازنده کلاس **BufferedWriter** را صدا زده ایم.
۳. بعد بجای پارامتر سازنده کلاس **BufferedWriter** که از نوع کلاس **Writer** هستش یک شی از کلاس **FileWriter** که یکی از فرزندان کلاس **Writer** هستش ایجاد کرده ایم
۴. در پایان بجای پارامتر سازنده کلاس **FileWriter** ، آدرس محلی که فایل قراره در آنجا ایجاد شود قرار می دهیم.

```
bw.write("www.javapro.ir");
bw.close();
```

- در اینجا با استفاده از متد **writer** متن مورد نظر را درون فایل نوشتیم.
- و در پایان هم با متد **close** این فایل را می بندیم.

void write(char[] array, int start, int len)

- اگر قصد داشتیم بخش های از داده های موجود در یک آرایه که از نوع کاراکتر هستند را در فایل بنویسیم از این متد استفاده می کنیم.
- پارامتر **start** نقطه شروع بازه ای از آرایه است که قراره در فایل ذخیره شود.
- پارامتر **len** تعداد کاراکترهایی است که از نقطه **start** به بعد قراره در فایل ذخیره شوند.
- ❖ طبق فرمول زیر محدوده ای از آرایه که قراره در فایل ذخیره شود بصورت زیر است:

[start, start+len-1]

مثلا اگر یک آرایه از نوع کاراکتر بصورت زیر داشته باشیم:

```
char ch[]={ 'j', 'a', 'v', 'a', 'l', 'i', 'k', 'e' };
```

و بصورت زیر این آرایه را در فایل ذخیره کنیم:

```
write(ch, 2, 5);
```

بخشی از آرایه `ch` که در فایل ذخیره می شود بصورت زیر است:

valik

در اینجا پارامترهای `start=2` و `len=5` می باشند:

پس بازه ای از آرایه که قراره در فایل ذخیره شود طبق فرمول `[start, start+len-1]` بصورت زیر می باشد:

[2,6]

حالا تمام کاراکترهایی که در محدوده خانه ۲ تا ۶ آرایه قرار می گیرند در فایل ذخیره می شوند.

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

- رنگ قرمز نقطه آغاز محدوده ما را نشان می دهد.

- رنگ سبز محدوده ای از آرایه که قراره در فایل ذخیره شود را نشان می دهد.

بجای فرمول می تونید `len` تا از پارامتر `start` را بشمارید!!! و به سمت راست آرایه حرکت کنید، از این طریق نیز محدوده سبز به دست می آید. فقط توجه داشته باشید این شمردن شامل ابتدای بازه یعنی پارامتر `start` نیز شود.

شمردن `len` تا از `start` به بعد را در زیر مشاهده می کنید:

```
start=2;
```

```
len=5;
```

۱.

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

۲.

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

۳.

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

.۴

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

.۵

0	1	2	3	4	5	6	7
j	a	v	a	l	i	k	e

این متد چقدر توضیحش طولانی شد!!!! خیلی هم کاربردی نداره!!!! تمام تلاشم کردم مفهومش رو روشن کنم! اگه باز گنگ بود بی خیال متد write قبلی رو بچسبید 😊

دیگه زشته خیلی توضیح دادیم یک مثالی هم ازش بزنیم 😊

مثال: اگه مثل من از توضیح زیاد خوشتون نییاد راحت از مثال زیر مفهوم رو یاد میگیرید!!!

```
package www.javapro.ir;

import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

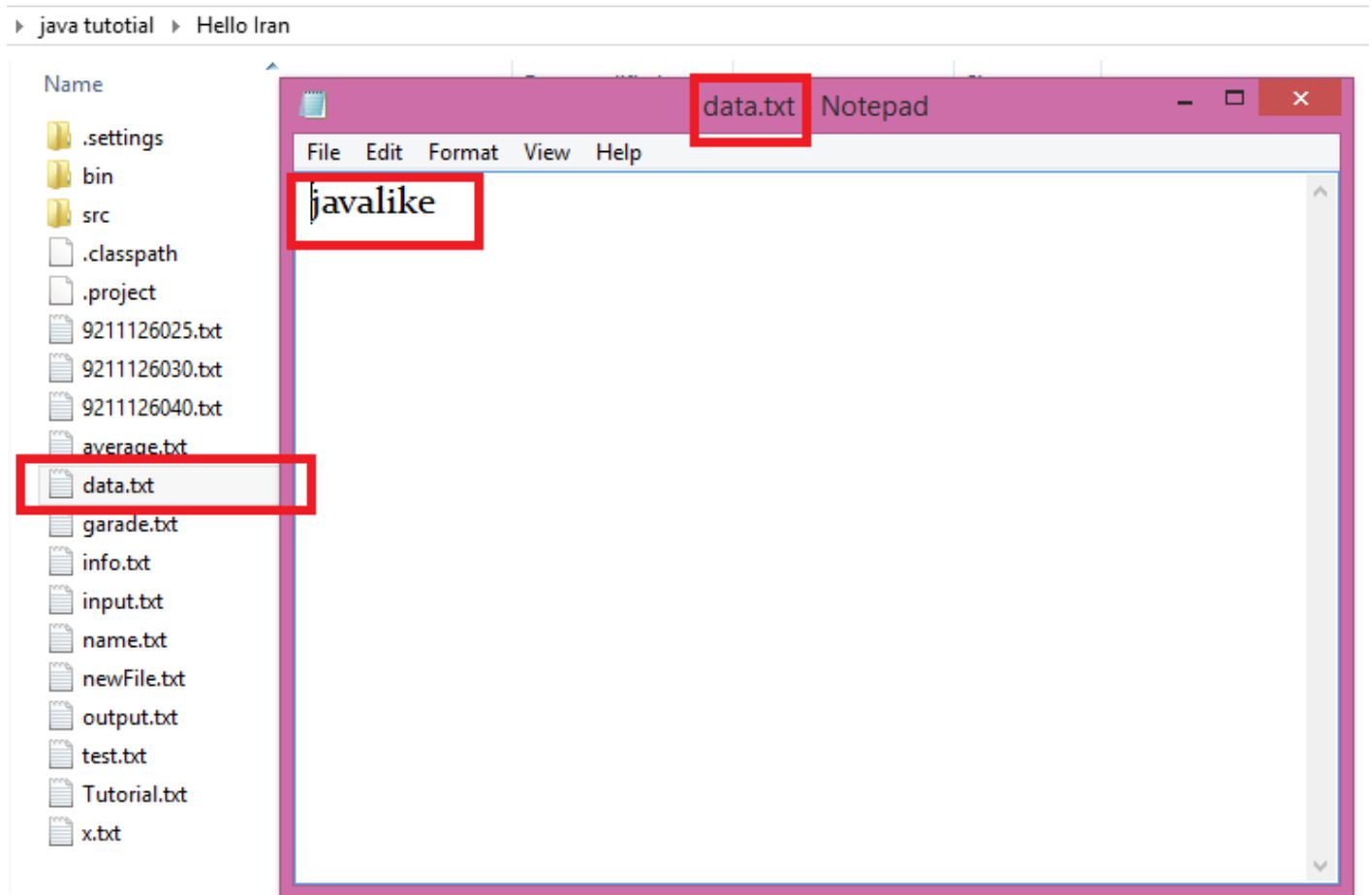
public class TestBufferedWriter {

    public static void main(String[] args) {
        try {
            BufferedWriter bw=new BufferedWriter(new FileWriter("data.txt"));
            char ch[]={ 'j', 'a', 'v', 'a', 'l', 'i', 'k', 'e' };
            bw.write(ch,0,8);
            bw.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }

    }

}
```

خروجی: خروجی این برنامه بصورت یک فایل با نام و فرمت data.txt می باشد. تصویر(۳)



تصویر (۳)

توضیحات:

```
BufferedWriter bw=new BufferedWriter(new FileWriter("data.txt"));
```

- این خط کد در مثال قبل توضیح دادم. برای ایجاد و نوشتن فایل ازش استفاده می کنیم.

```
char ch[]={ 'j', 'a', 'v', 'a', 'l', 'i', 'k', 'e' };
```

- یک آرایه از کاراکتر تعریف کرده و مستقیم مقدار دهی اولیه کرده ایم.

```
bw.write(ch,0,8);
```

- در اینجا $start=0$ و $len=8$ می باشد، طبق فرمول $[start, start+len-1]$ بازه بخشی از آرایه ch که قراره در فایل ذخیره شود بصورت زیر است:

[0,7]

پس تمام کاراکترهای آرایه که در بازه ۰ تا ۷ قرار می گیرد در فایل ذخیره می شوند.

```
bw.close();
```

- در پایان با استفاده از این متد فایل خود را می بندیم.

void newLine()

هنگام نوشتن داده ها در فایل اگر از این متد استفاده کنیم داده ها از سطر جدید نوشته می شوند. نوعی جداکننده سطرها از یکدیگر هستند. اگر بدون استفاده از این متد داده ها رو داخل فایل ذخیره کنیم همه داده ها در یک سطر از فایل قرار می گیرند، اما به هر دلیلی خواستیم داده ها رو در سطرهای مجزا ذخیره کنیم از این متد استفاده می شود، مثلا اگر قصد داشتیم اطلاعات یک دانشجو رو در فایل ذخیره کنیم به گونه ای که هر ویژگی دانشجو مثل نام و نام خانوادگی در سطر اول، نمرات در سطر دوم، شماره دانشجویی در سطر سوم و... از این متد استفاده می کنیم. چیزی شبیه دکمه Enter در کیبورد هستش که وقتی میزنیم نشانه گر به سطر جدید می رود و ما می توانیم از سطر جدید شروع به نوشتن کنیم اینم همین طور هست.

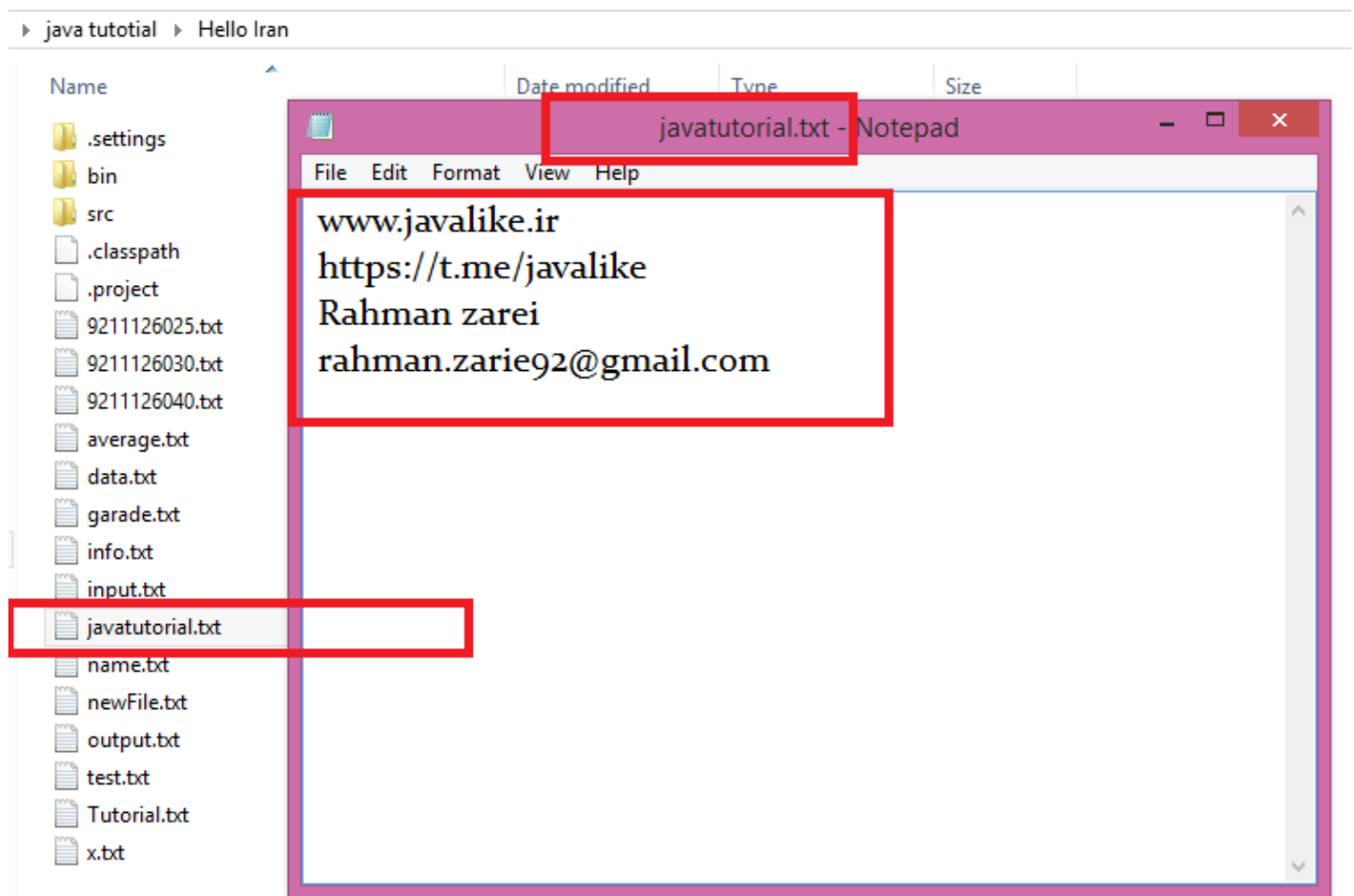
مثال :

```
package www.javapro.ir;
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class TestBufferedWriter {

    public static void main(String[] args) {
        try {
            BufferedWriter bw = new BufferedWriter(new
FileWriter("javatutorial.txt"));
            bw.write("www.javalike.ir");
            bw.newLine();
            bw.write("https://t.me/javalike");
            bw.newLine();
            bw.write("Rahman zarei");
            bw.newLine();
            bw.write("rahman.zarie92@gmail.com");
            bw.close();
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }
}
```

خروجی: خروجی این برنامه بصورت یک فایل با نام و فرمت javatutorial.txt می باشد. تصویر (۴)



تصویر (۴)

توضیحات:

```
bw.write("www.javalike.ir");  
bw.newLine();  
bw.write("https://t.me/javalike");  
bw.newLine();  
bw.write("Rahman zarei");  
bw.newLine();  
bw.write("rahman.zarie92@gmail.com");  
bw.close();
```

• با متد write متن مورد نظر را درون فایل ریختیم.

- با استفاده از متد `newline()` به سطر بعد می رویم و همان طور که در تصویر ۴ مشاهده می کنید بعد از نوشتن متن [www.javajlike.ir](https://t.me/javajlike) به سطر بعد رفته و متن <https://t.me/javajlike> نوشته ایم و باز به سطر بعد رفته و....
- پس اگر قصد داشتیم سطر به سطر اطلاعات خودمون رو در فایل ذخیره کنیم از متد `newline` استفاده می کنیم.

این بخش از آموزش نیز تمام شد امیدوارم که مفاهیم رو واضح و روشن گفته باشم. نیاز به حل مثال بیشتر برای یادگیری دارید، خیلی توضیحات مهم نیست فقط حل مکرر مثال و تمرین چاره یادگیری است. سعی می کنم نمونه مثال تهیه کنم و در کانال و سایت قرار بدم.

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.