

برای نوشتن در فایل از کلاس FileWriter استفاده میکنیم. برای این منظور باید ابتدا یک آبجکت از این کلاس تعریف کنیم.

FileWriter writer = new FileWriter(Path);  
 نکته: آبجکت writer نام آبجکت از کلاس FileWriter برای نوشتن در فایل است و Path جایگزین writerName نام آبجکت در بلاک try...catch نوشته میشود و در نظر بگیرید که فرمتی که میخواهید بنویسید باید به صورت String باشد.

پس از متوجه شدن نحوه کلاس FileWriter:

نوشتن در فایل با استفاده از متود write:

writerName.write("string\_content");

جایگزین content این متود و محتوای متنی که میخواهید در فایل بنویسید است.

نکته ۱: با اجرای این دستور محتوای قبلی فایل پاک شده و محتوای جدید در آن نوشته میشود.

نکته ۲: برای پایان دادن به نوشتن در فایل باید حتماً آبجکت را با متود زیر فراموشی کنیم.

writerName.close();

کنیم.

اگر بخواهیم در فایل را با متود زیر فراموشی کنیم باید متود زیر را بنویسیم:

احتمال فراموشی



2019

شهریور ۹۸

Sep 19



پنجشنبه

۱۹ محرم

۱۴۰۱

مثال: ~~نویسنده فایل~~  
String FilePath = "/home/majtaba/TestFolder/TestFile.txt";

try {

FileWriter myWriter = new FileWriter(FilePath);

myWriter.write("Hello, world!");

myWriter.close();

System.out.println("successfully wrote to the file.");

}

catch (IOException e) {

System.out.println("Error: " + e.getMessage());

}

نکته: در این مثال، متد write() را به کار می‌بریم که علاوه بر نوشتن متن، فایل را هم می‌سازد. اگر فایل وجود داشته باشد، متن جدید را به انتهای آن اضافه می‌کند. اگر می‌خواهیم فایل را به صورت کامل بازنویسی کنیم، باید متد write() را با پارامتر true فراخوانی کنیم.

FileWriter writeName = new FileWriter(FilePath, true);



## خواندن از یک فایل متنی

برای خواندن از فایل متنی از کلاس Scanner استفاده می‌کنیم.

```
File fileObjectName = new File (FilePath);
```

```
Scanner myReaderName = new Scanner (fileObjectName);
```

متد `hasNextLine` از آبجکت `Scanner` : برابر پرسش وجود متن در خط جاری از فایل مشخص شده.

```
ReaderName.hasNextLine();
```

اگر خط وجود داشته باشد `true` و در غیر این صورت `false` برمی‌گرداند.

متد `nextLine` از آبجکت `Scanner` : برابر خواندن خط جاری و تغییر اشاره‌گر خواننده به خط بعدی فایل متنی.

```
ReaderName.nextLine();
```



2019

شهریور ۹۸

Sep 22

۱۴۴۱

یکشنبه

۲۲ محرم

مثال: خواندن فایل متنی به صورت خط به خط و چاپ آن در خروجی

```
String FilePath = "~/home/mojtaba/TestFolder/TestFile.txt";
```

```
try {
```

```
File myFile = new File(FilePath);
```

```
Scanner myReader = new Scanner(myFile);
```

```
while (myReader.hasNextLine())
```

```
{
```

```
String data = myReader.nextLine();
```

```
System.out.println(data);
```

```
}
```

```
myReader.close();
```

```
}
```

```
catch (Exception e)
```

```
{
```

```
System.out.println(e.getMessage());
```

```
}
```

نکته: از آنجا که اسکینر Scanner بر خواندن از فایل استوار است و یک شیء است، باید خط تفریق را بگذاشت Scanner در بابت try...catch قرار گیرد



2019

مہر ۹۸

Sep 24

۱۴۴۱

سہ شنبہ

۲۴ محرم

تمرین ۱: ذخیرہ اطلاعات دفترچه تلفن:

برنامہ ایف بی سیڈ کہ نام (Name)، نام خانوادہ (Family)، آکریس (Address) و شمار تلفن (Tel #) یک مخاطب را از کاربیر دریافت و آن را در کتاب حذف ذخیرہ

Name, Family, Address, Tel #

در هر خط فایلی می نویسد.

تمرین ۲: حذف اطلاعات دفترچه تلفن:

برنامہ ایف بی سیڈ کہ نام و نام خانوادہ یک مخاطب را به عنوان ورودی دریافت می کند و در صورت وجود چنین مخاطبی، اطلاعات او را از فایلی دفترچه تلفن حذف می کند.

تمرین ۳: ویرایش اطلاعات دفترچه تلفن:

برنامہ ایف بی سیڈ کہ به طور مناسب نام و نام خانوادہ مخاطب را به همراه فیلدهای ویرایشی (آکریس یا شماره تلفن) از ورودی دریافت می کند و فایلی دفترچه تلفن را مناسب بارگذاری و ویرایش می کند.

تمرین ۴: جست و جوی دفترچه تلفن:

نام و نام خانوادہ مخاطب را دریافت می کند، شماره تلفن و آکریس او را به نام