

Ch10

파일 입출력

객체지향프로그래밍(기본) 2019
경상대학교 항공우주및소프트웨어공학전공

10.1 입출력 스트림

파일에 데이터를 저장 할 수 있고 파일에 있는 데이터를 읽어 프로그램에 적용 할 수 있다.

컴퓨터의 데이터는 스트림 형태로 이동한다. 먼저 들어간 데이터가 먼저 나오는 순차적인 선입 선출의 구조로 되어 있으며 대부분의 데이터는 단방향으로 진행되기 때문에 읽기와 쓰기를 동시에 할 수 없다.

스트림은 출발지 혹은 목적지로의 연결을 나타낼 수 있고 스트림을 사용하기 위해 먼저 열어야 하며 사용 후는 닫는다.


10.2 파일의 생성 및 저장

입력과 출력의 스트림은 InputStream과 OutputStream을 기반으로 생성된다.

파일 입출력을 위한 스트림으로 FileInputStream과 FileOutputStream가 있다.

```
import java.io.FileOutputStream;
```

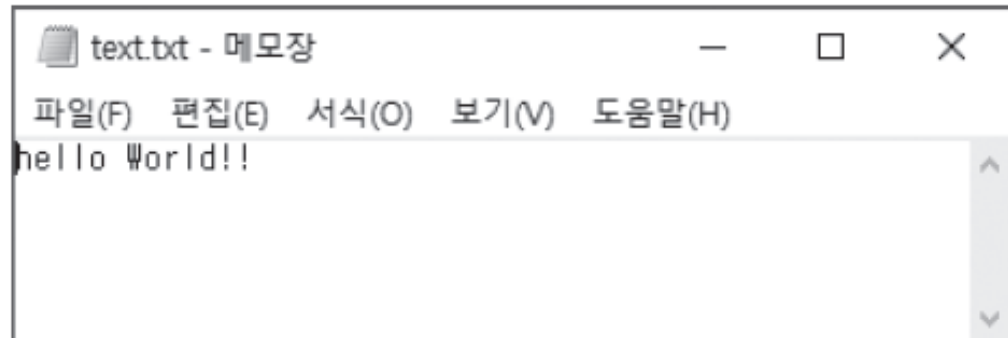
```
try {  
    FileOutputStream file = new FileOutputStream("text.txt");  
    file.close();  
}catch(IOException e) {
```

내용을 저장할 파일 지정 혹은
파일이 없을때 파일 생성 

파일 입력

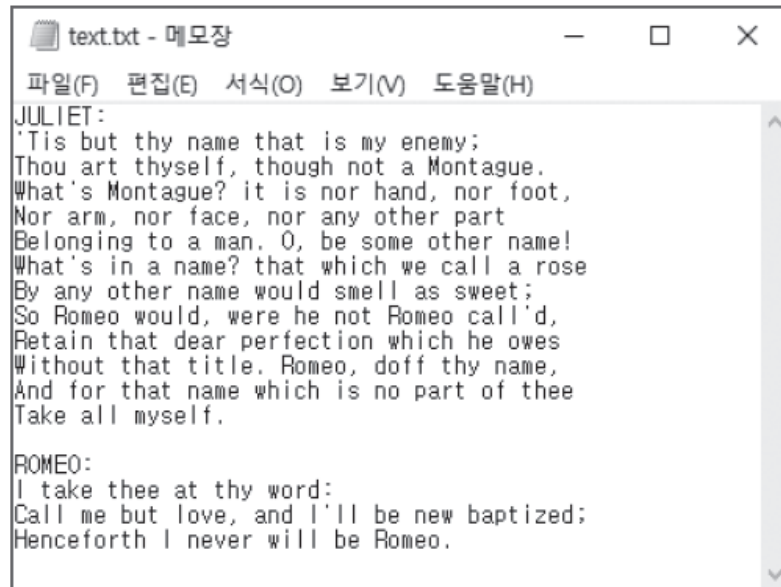
```
try {  
    FileOutputStream file = new FileOutputStream("text.txt");  
  
    file.write("hello World!!".getBytes());  
  
    file.close();  
}catch(IOException e) {  
    System.out.println(e.getMessage());  
}
```

↑ *FileOutputStream에서는 바이트 단위로 일기 때문에
바이트 값을 가져온다.*



파일 연결하여 입력

```
FileOutputStream file = new FileOutputStream("text.txt");  
-> FileOutputStream file = new FileOutputStream("text.txt", true);
```



다양한 객체를 이용한 입력

FileWriter를 이용한 입력

```
FileWriter file = new FileWriter("text.txt");  
-> FileWriter file = new FileWriter("text.txt", true);
```

PrintWriter를 이용한 입력

```
PrintWriter file = new PrintWriter("text.txt");  
-> PrintWriter file = new PrintWriter(new FileWriter("text.txt",true ));
```

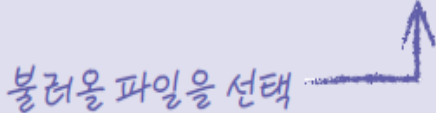
10.3 파일의 출력

파일의 내용을 화면에 출력하는데 `FileInputStream` 객체가 사용된다.

```
byte[] b = new byte[1024];
FileInputStream file = new FileInputStream("text.txt");
file.read(b);
System.out.println(new String(b));

file.close();
```

불러올 파일을 선택



없는 파일의 출력

읽고자 하는 파일이 패스 폴더에 존재하지 않을 때 오류가 발생한다.

```
FileInputStream file = new FileInputStream("no.txt");
```

결과

no.txt (지정된 파일을 찾을 수 없습니다)

없는 파일의 출력

읽고자 하는 파일이 패스 폴더에 존재하지 않을 때 오류가 발생한다.

```
FileInputStream file = new FileInputStream("no.txt");
```

결과

no.txt (지정된 파일을 찾을 수 없습니다)

버퍼 단위 파일 출력

버퍼단위로 파일을 읽어 화면에 출력하기 위해 `BufferedReader`와 `FileReader`를 사용한다.

`FileReader`로 지정된 파일의 입력 스트림을 생성하고 `BufferedReader`로 파일의 크기만큼의 버퍼를 가진 객체를 생성하여 `BufferedReader`객체의 `readLine` 메소드를 이용하여 한 줄씩 읽는다.

`readLine` 메소드는 더 이상 읽을 라인이 없을 경우 `null`값을 반환한다.

버퍼 단위 파일 출력

```
BufferedReader file = new BufferedReader(new FileReader("text.txt"));

while(true){
    String str = file.readLine();

    if( str == null )
        break;

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

10.4 Date을 이용한 Log파일 만들기

로그(Log)파일은 컴퓨터의 프로그램이 실행 중에 발생하는 이벤트나 각종 통신 메시지를 기록한 파일이다.

로그파일을 생성할 때 이벤트나 발생이나 메시지 전달 시간의 기록 또한 저장된다.

시간의 정보를 받기 위해 Date 클래스를 이용하며 받은 시간정보를 표현하기 위해 SimpleDateFormat 클래스를 사용한다.

SimpleDateFormat

Date and Time Pattern	Result
"yyyy.MM.dd G 'at' HH:mm:ss z"	2001.07.04 AD at 12:08:56 PDT
"EEE, MMM d, ''yy"	Wed, Jul 4, '01
"h:mm a"	12:08 PM
"hh 'o'clock' a, zzzz"	12 o'clock PM, Pacific Daylight Time
"K:mm a, z"	0:08 PM, PDT
"yyyyy.MMMMM.dd GGG hh:mm aaa"	02001.July.04 AD 12:08 PM
"EEE, d MMM yyyy HH:mm:ss Z"	Wed, 4 Jul 2001 12:08:56 -0700
"yyMMddHHmmssZ"	010704120856-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSZ"	2001-07-04T12:08:56.235-0700
"yyyy-MM-dd'T'HH:mm:ss.SSSXXX"	2001-07-04T12:08:56.235-07:00
"YYYY-'W'ww-u"	2001-W27-3

Date and Time Patterns

Letter	Date or Time Component	Presentation	Examples
G	Era designator	Text	AD
y	Year	Year	1996; 96
Y	Week year	Year	2009; 09
M	Month in year	Month	July; Jul; 07
w	Week in year	Number	27
W	Week in month	Number	2
D	Day in year	Number	189
d	Day in month	Number	10
F	Day of week in month	Number	2
E	Day name in week	Text	Tuesday; Tue
u	Day number of week (1 = Monday, ..., 7 = Sunday)	Number	1
a	Am/pm marker	Text	PM
H	Hour in day (0-23)	Number	0
k	Hour in day (1-24)	Number	24
K	Hour in am/pm (0-11)	Number	0
h	Hour in am/pm (1-12)	Number	12
m	Minute in hour	Number	30

Letter	Date or Time Component	Presentation	Examples
s	Second in minute	Number	55
S	Millisecond	Number	978
z	Time zone	General time zone	Pacific Standard Time; PST; GMT-08:00
Z	Time zone	RFC 822 time zone	-0800
X	Time zone	ISO 8601 time zone	-08; -0800; -08:00

실행 시간 Log 파일

```
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
try {
    FileOutputStream file = new FileOutputStream("TimeLog.txt");

    file.write(format.format(date).getBytes());

    file.close();
}catch(IOException e) {
    System.out.println(e.getMessage());
}
```

