

제 45강

입출력, 자바의 IO 패키지

교재: p258~262

목차

1. 입출력, 자바의 IO 패키지

1. IO와 스트림(Stream)
2. 바이트 기반 스트림
3. 바이트 기반 보조 스트림

1. IO와 스트림(Stream)

1) IO?

: Input(입력), Output(출력)

입력: 프로그램에 데이터가 들어오는 것

출력: 프로그램의 데이터가 나가는 것

2) 스트림?

: 데이터의 흐름



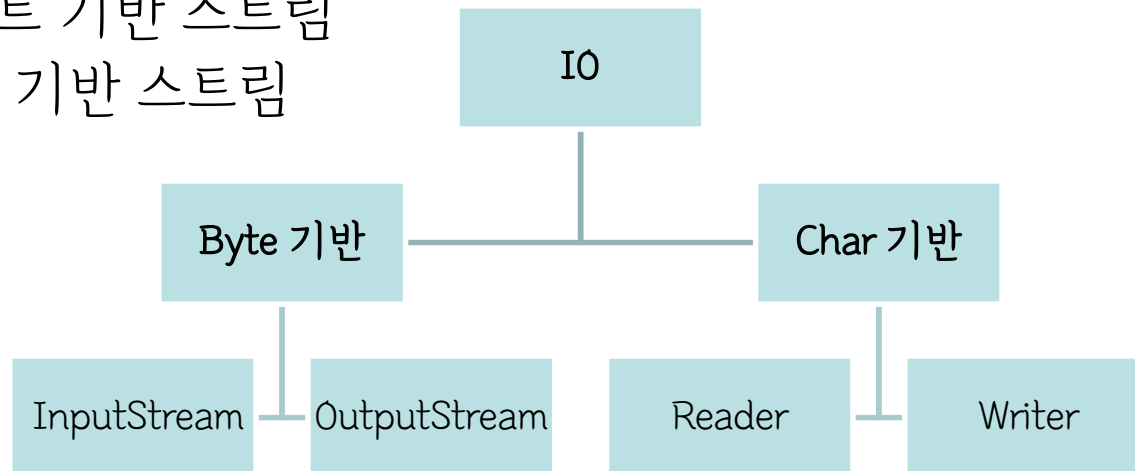
1. IO와 스트림(Stream)

2) 스트림

스트림의 특징

- 단 방향이다.
- FIFO 구조를 갖고 있다.
- 출력단위에 따라 바이트 기반과 문자단위로 분류된다.

> 출력단위: 1byte-> 바이트 기반 스트림
 문자 -> 문자 기반 스트림



2. 바이트 기반 스트림

바이트 기반 스트림?

: 바이트 단위로 데이터를 입출력하는 스트림

[1] 바이트 기반 스트림의 활용

InputStream	OutputStream
abstract int read()	abstract void write(int b)
int read(byte[] b)	void write(byte []b)
int read(byte []b,int off,int len)	void write(byte []b ,int off,int len)

→ 입력과 출력에 따라, read() 와 write()를 구현하여 사용한다.

2. 바이트 기반 스트림

바이트 기반 스트림?

: 바이트 단위로 데이터를 입출력하는 스트림

[1] 바이트 기반 스트림의 활용

입력 스트림	출력 스트림	대상
FileInputStream	FileOutputStream	파일
PipedInputStream	PipedOutputStream	메모리
AudioInputStream	AudioOutputStream	오디오
ByteArrayInputStream	ByteArrayOutputStream	프로세스

데이터를 읽거나 쓰는 대상에 따라서 적절한 스트림을 사용한다.

2. 바이트 기반 스트림

[1] 바이트 기반 스트림의 활용

스트림을 사용 할 때는, 인스턴스 생성시 적절한 값을 전달한다.

ex) 파일 입출력

```
FileInputStream fis= null;  
fis = new FileInputStream("파일위치");  
fis.read();
```

3. 바이트 보조 기반 스트림

“바이트 기반 보조스트림은 실제 바이트 기반 스트림의 성능을 향상시키는 역할”

[1] 바이트 기반 보조스트림의 활용

입력보조스트림	출력보조스트림	사용
FilterInputStream	FileterOutputStream	필터를 이용한 입출력
BufferedInputStream	BufferedOutputStream	버퍼를 통한 입출력
DataInputjStream	DataOutputStream	기본 단위로 데이터 처리
x	PrintStream	print,printf, println 사용

ex) BufferedInputStream 활용

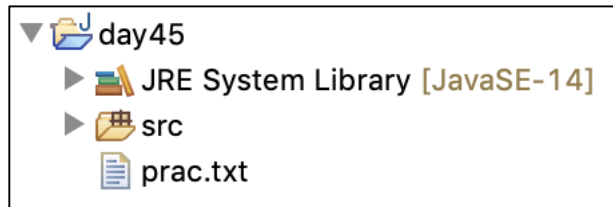
```
FileInputStream fis=null;
fis=new FileInputStream("파일 위치");
BufferedInputStream bis=new BufferedInputStream(fis);
```


3. 바이트 보조 기반 스트림

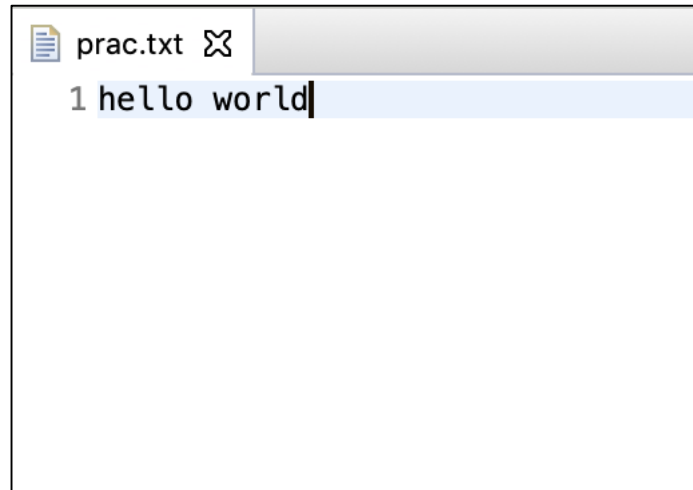
<실습> Exam-100.java

바이트 기반 스트림과 보조스트림을 이용하여 파일을 복사하는 코드 작성하기

[prac.txt 파일 생성]



경로: day45 프로젝트 내



3. 바이트 보조 기반 스트림

<실습> Exam-100.java

바이트 기반 스트림과 보조스트림을 이용하여 파일을 복사하는 코드 작성하기

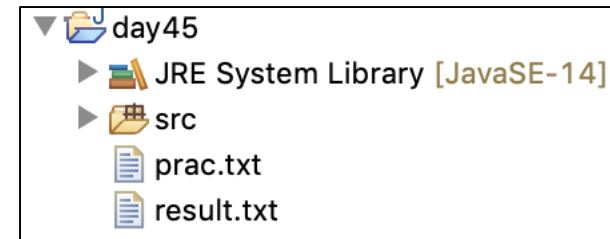
[실행 결과]

```
import java.io.*;

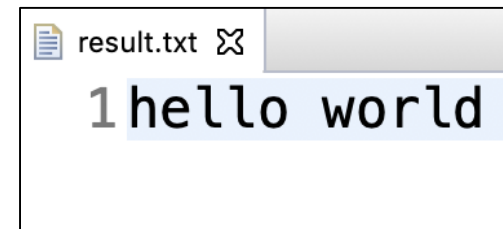
public class File01 {
    public static void main(String[] args) throws IOException{
        FileInputStream fis=null;
        FileOutputStream fos= null;
        fis=new FileInputStream("prac.txt");
        fos=new FileOutputStream("result.txt");

        BufferedInputStream bis=new BufferedInputStream(fis);
        BufferedOutputStream bos= new BufferedOutputStream(fos);
        int data;
        while((data=fis.read())!=-1) {
            bos.write(data);
        }
        bos.close();
        bis.close();
    }
}
```

입출력에 관련한 예외 처리



1) result.txt 파일 생성



2) prac.txt 파일 내용 복사