

제 13 장

입출력 스트림과 통신

Part-1: Input/Output Stream



입출력 스트림(Stream)

□ 스트림

- ▣ 정의: 순서가 존재하는 데이터의 연속적 흐름(혹은 sequence)
- ▣ 데이터흐름을 관장하는 입출력 소프트웨어 컴포넌트

□ 자바의 스트림

- ▣ 자바의 스트림은 입출력 장치와 프로그램을 연결하며, 이들 사이의 데이터 전송을 스트림으로 처리하는 소프트웨어 컴포넌트

▣ 입력 스트림

- ▣ 입력 장치로부터 자바 프로그램으로 전달되는 데이터의 흐름 혹은 데이터 전송 소프트웨어 모듈

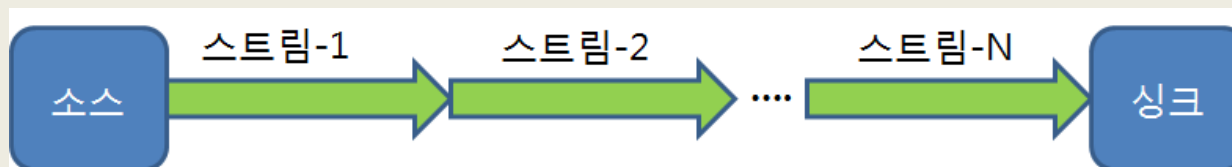
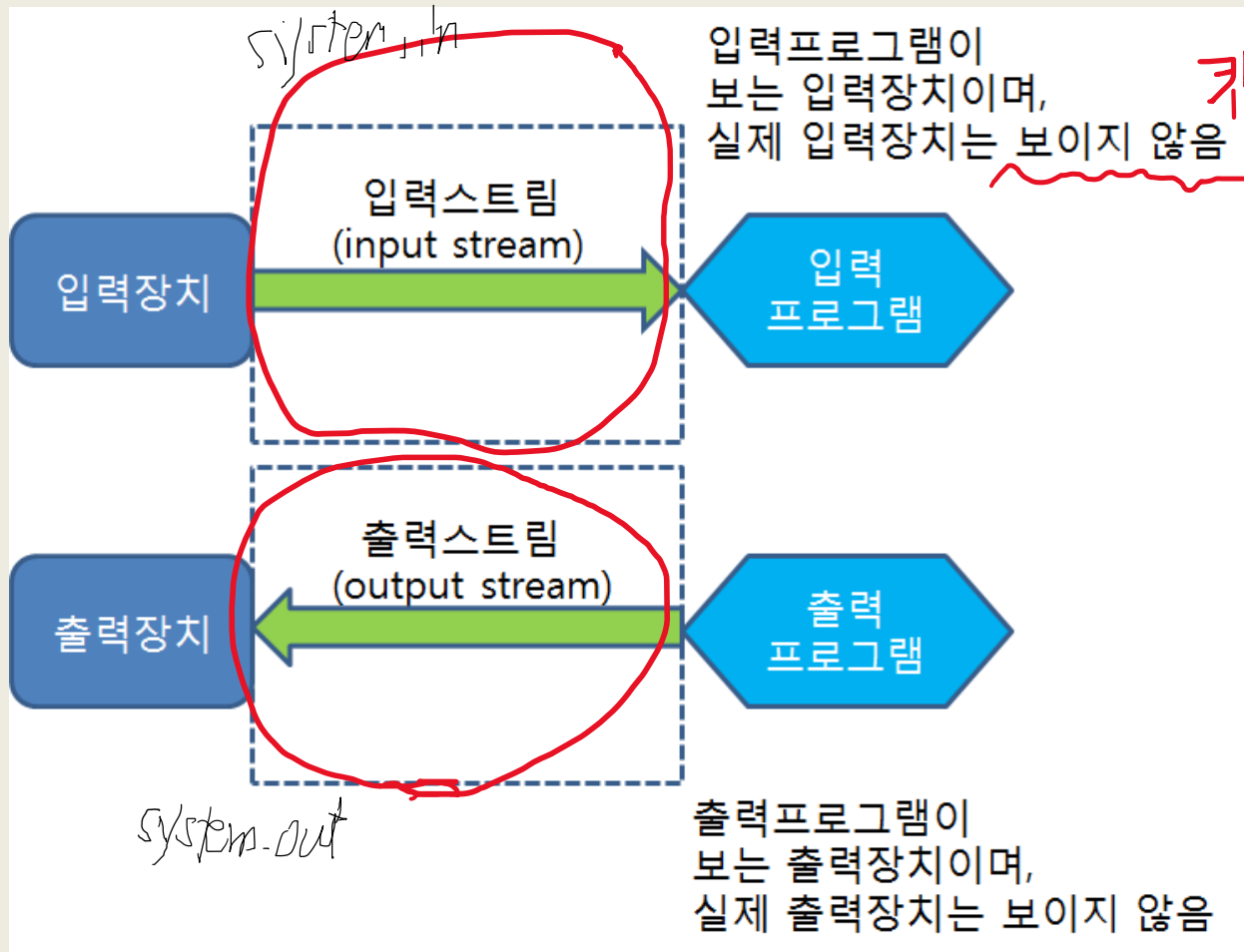
▣ 출력 스트림

- ▣ 자바 프로그램에서 출력 장치로 보내는 데이터의 흐름 혹은 데이터 전송 소프트웨어 모듈

□ 입출력 스트림 기본 단위 : 바이트 (Byte Stream)

□ 자바 입출력 스트림 특징

- ▣ 단방향 스트림, 선입선출(FIFO) 구조
- ▣ 2개 이상의 스트림이 연결될 수 있다





입출력 스트림 종류

□ 바이트 스트림과 문자 스트림

■ 바이트 스트림

- 가공을 거치지 않은 원자료의 단순 바이트 데이터의 흐름
- 예) 이진 파일을 읽는 입력 스트림

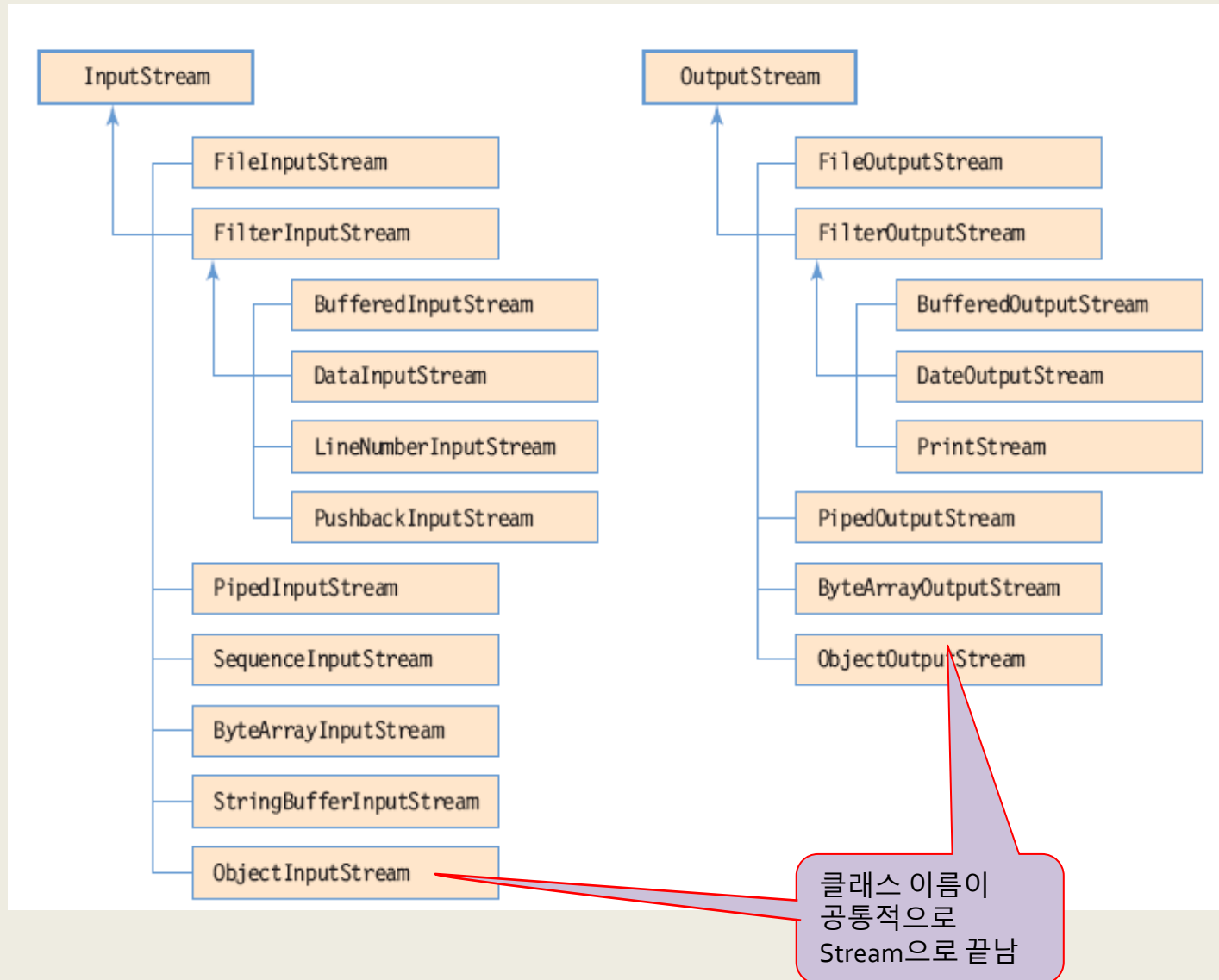
■ 문자 스트림

- 문자의 흐름으로 처리.
- 문자가 아닌 이진 데이터는 문자스트림에서 처리하지 못함
- 예) 텍스트 파일을 읽는 입력 스트림

□ JDK는 입출력 스트림을 구현한 다양한 클래스 제공

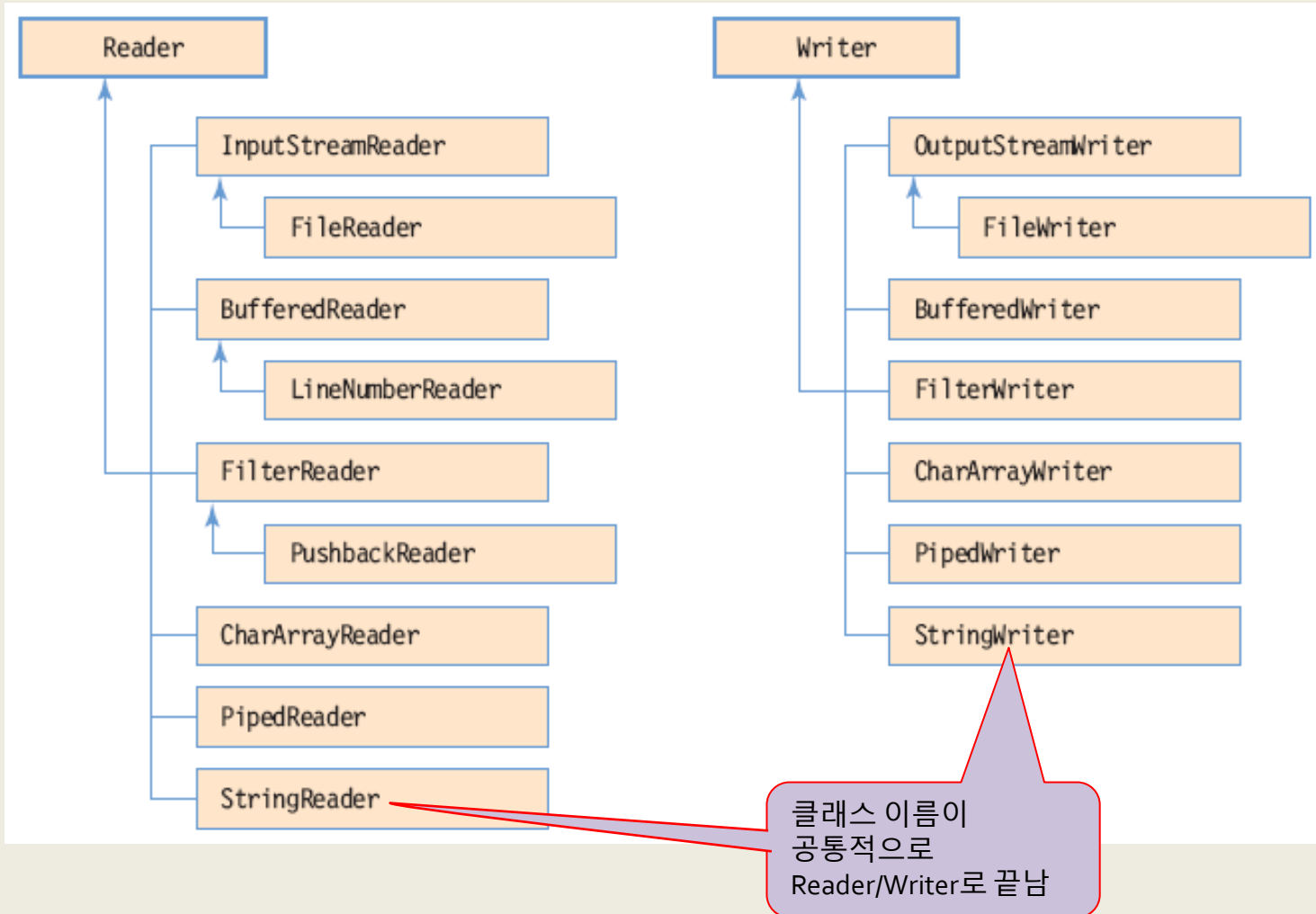


바이트 스트림 클래스 계층 구조





문자 스트림 클래스 계층 구조



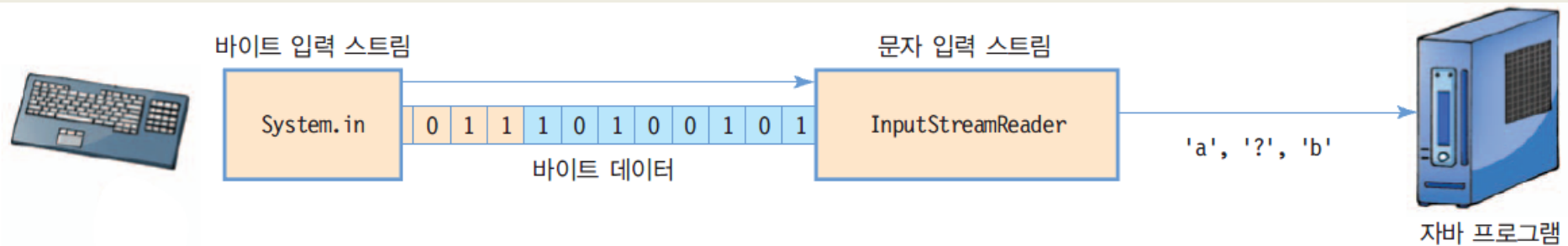


기본적인 메소드

- ❑ InputStream 추상 클래스
 - abstract int read() - 한 바이트를 읽어서 반환한다(0에서 255 사이의 정수).
- ❑ OutputStream 추상 클래스
 - abstract void write(int b) - 한 바이트를 특정한 장치에 쓴다.
- ❑ Reader 추상 클래스
 - abstract int read() - 한 문자를 읽어서 반환한다.
- ❑ Writer 추상 클래스
 - abstract void write(int c) - 한 문자를 특정한 장치에 쓴다.

스트림은 연결될 수 있다

* 표준 입력 스트림 System.in에 InputStreamReader 스트림을 연결하는 사례

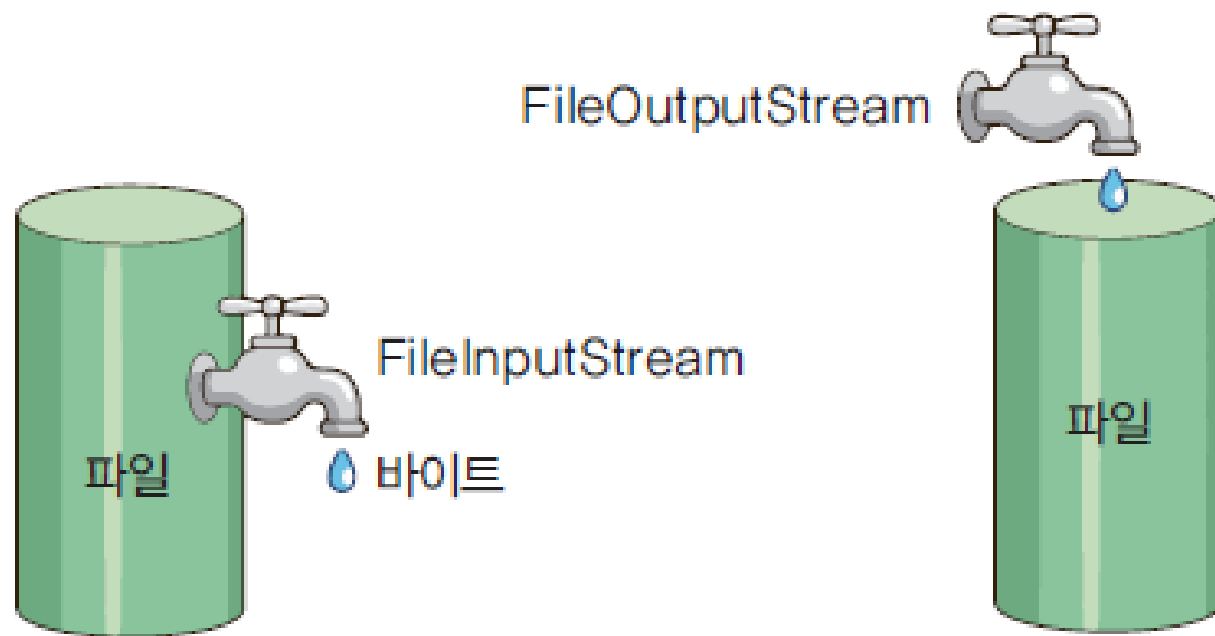


```
InputStreamReader rd = new InputStreamReader(System.in);  
int c = rd.read(); // 키보드에서 문자 읽음
```




바이트 스트림

- ❑ 바이트 스트림
 - ▣ 바이트 단위의 2진 값 흐름을 읽고 저장하는 컴포넌트
- ❑ 바이트 스트림 클래스
 - ▣ java.io 패키지에 포함
 - ▣ InputStream/OutputStream
 - ▣ 추상 클래스
 - ▣ 바이트 입출력 스트림을 다루는 모든 클래스의 수퍼 클래스
 - ▣ FileInputStream/FileOutputStream
 - ▣ 파일로부터 바이트 단위로 읽거나 저장하는 클래스
 - ▣ 2진 파일의 입출력
 - ▣ DataInputStream/DataOutputStream
 - ▣ 자바의 기본 데이터 타입의 값(변수)을 바이너리 값 그대로 입출력
 - ▣ 문자열도 바이너리 형태로 입출력



파일 입출력 바이트 스트림



❑ 표준입력 장치와의 송수신

- ❑ InputStream은 추상객체이므로, 자체 생성이 불가능
 - ❑ 생성가능한 다른 스트림과 연결 사용
- ❑ System.in
 - ❑ 표준입력장치인 키보드로부터의 입력
 - ❑ 문자로 변환되어 사용되는 것이 일반적
- ❑ 문자스트림 객체인 **InputStreamReader**와 연결하여 문자스트림 객체의 메소드를 이용
- ❑ 예) 키보드로부터의 입력

```
InputStreamReader kbd_isr = new InputStreamReader(System.in) ;
```

```
kbd_isr.METHOD_NAME(...);
```



❑ 표준 출력 장치와의 송수신

- ▣ PrintStream은 자체 생성자를 가지고 생성이 가능
- ▣ 다양한 메소드들을 지원하고 있으므로 직접 PrintStream 객체가 제공하는 메소드를 사용하는 것이 일반적이다.

```
PrintStream screen = System.out ;
```

```
        혹은 new PrintStream((OutputStream)System.out))
```

```
screen.METHOD_NAME(...) ; 혹은 System.out.METHOD_NAME(...) ;
```

```
예를 들면, screen.println() ;
```



FileInputStream을 이용한 파일 읽기

□ 파일 전체를 읽어 화면에 출력하는 코드 샘플

```
FileInputStream fin = new FileInputStream("c:\\test.txt");
```

입력 바이트 스트림객체를 생성하고
C:\test.txt 파일 오픈 (수도꼭지를 연다)

```
int c;
```

```
while((c = fin.read()) != -1) {
```

파일 끝까지 반복하며 한 바이트씩 c에 읽어 들임.
파일의 끝을 만나면 read()는 -1 리턴

```
    System.out.print((char)c);
```

바이트 c를 문자로 변환하여 화면에 출력

```
}
```

```
fin.close();
```

스트림을 닫음. 파일도 닫힘 (수도꼭지 잠금).
더 이상 스트림으로부터 읽을 수 없음



예제 : 윈도우즈에 있는 system.ini 파일을 읽어 화면에 출력하기

FileInputStream을 이용하여 사용자 컴퓨터의 windows 디렉터리에 있는 system.ini 파일을 읽고 화면에 출력하라. system.ini 파일은 텍스트 파일이다.

```
import java.io.*;

public class FileInputStreamEx {
    public static void main(String[] args) {
        FileInputStream fis = null;
        try {
            fis = new FileInputStream("c:\\windows\\system.ini");

            int c;
            while ((c = fis.read()) != -1) {
                System.out.print((char)c);
            }
            fis.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```

키보드에서 ctrl-z를 누르면
입력 종료

```
; for 16-bit app support
[386Enh]
woafont=dosapp.fon
EGA8oWOA.FON=EGA8oWOA.FON
EGA4oWOA.FON=EGA4oWOA.FON
CGA8oWOA.FON=CGA8oWOA.FON
CGA4oWOA.FON=CGA4oWOA.FON
```

```
[drivers]
wave=mmdrv.dll
timer=timer.drv
```

```
[mci]
```



FileOutputStream을 이용한 파일 쓰기

□ 바이너리 값을 파일에 저장하는 바이트 스트림 코드

```
FileOutputStream fout = new FileOutputStream("c:\\test.out");
```

출력 바이트 스트림 객체를 생성하고 C:\test.out 파일 오픈

```
int num[]={1,4,-1,88,50};
```

```
byte b[]={7,51,3,4,1,24};
```

```
for(int i=0; i<num.length; i++)
```

```
    fout.write(num[i]);
```

파일에 정수 값(바이너리)을 그대로 기록

```
fout.write(b);
```

파일에 바이트 배열(바이너리) 값을 그대로 기록

```
fout.close();
```

스트림을 닫음. 파일도 닫힘. 더 이상 스트림으로부터 읽을 수 없음

파일에 있는 각 바이너리 값들은
문자 정보가 아님. 바이너리 값에
대응하는 그래픽 심볼들

00000000h: 01 04 FF 58 32 07 33 03 04 01 18

; ǀ X2 3 ǂ ǀ↑

test.out 파일의 내부

예제 : FileOutputStream을 이용한 파일 쓰기

정수 타입의 결과 값을
FileOutputStream을 이용하여
파일에 저장한다. 다시 이
파일에서 정수형 변수로 읽고
이전에 계산된 결과 값과
같은지 확인하라.

10 9 8 7 6 5 4 3 2 1

```
import java.io.IOException;
public class FileOutputStreamEx {
    public static void main(String[] args) {
        try {
            FileOutputStream fout = new FileOutputStream("C:\\WWtest.out");
            FileInputStream fin = null;

            for (int i=0; i<10; i++) {
                int n = 10-i; // 계산의 결과를 저장
                fout.write(n); // 파일에 결과값을 바이너리로 저장
            }
            fout.close(); //스트림을 닫는다.

            fin = new FileInputStream("C:\\WWtest.out");
            int c=0;
            while ((c = fin.read()) != -1) {
                System.out.print(c + " ");
            }
            fin.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```

10 9 8 7 6 5 4 3 2 1
00000000h: 0A 09 08 07 06 05 04 03 02 01 ; -|J L r

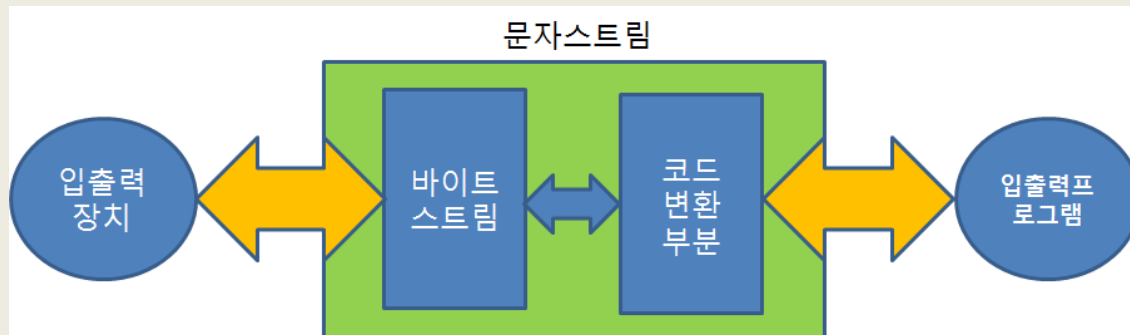


문자 스트림

- ❑ 문자 스트림
 - ▣ 유니 코드로 된 문자를 입출력 하는 스트림 컴포넌트
 - ▣ 문자로 표현되지 않는 데이터는 다루지 않음
 - ▣ 문자 스트림은 이미지, 동영상과 같은 바이너리 데이터는 입출력 할 수 없음 - 문자 스트림은 문자 데이터만 입출력 가능
- ❑ 문자 스트림을 다루는 클래스
 - ▣ Reader/Writer
 - ▣ Java.io 패키지에 포함
 - ▣ 추상 클래스 - 문자 스트림을 다루는 모든 클래스의 슈퍼 클래스
 - ▣ InputStreamReader/OutputStreamWriter
 - ▣ 바이트 스트림과 문자 스트림을 연결시켜주는 다리 역할
 - ▣ 지정된 문자집합 이용
 - ▣ InputStreamReader: 바이트를 읽어 문자로 인코딩
 - ▣ OutputStreamWriter: 문자를 바이트로 디코딩하여 출력
 - ▣ FileReader/FileWriter
 - ▣ 텍스트 파일에서 문자 데이터 입출력



- ❑ 문자스트림은 내부적으로 바이트스트림을 사용하여 구현
 - ❑ 일단은 원본 데이터 그대로 액세스 한 후에 필요한 코드로 변환하여 입출력이 일어난다
 - ❑ 문자스트림 클래스는 바이트스트림 클래스의 wrapper 클래스인 것이다





예제 : FileReader를 이용한 텍스트 파일 읽기 - system.ini 파일 읽기

FileReader를 이용하여 사용자 컴퓨터의 windows 디렉터리에 있는 system.ini 파일을 읽고 화면에 출력하라. system.ini 파일은 텍스트 파일이다.

```
import java.io.*;

public class FileReaderEx {
    public static void main(String[] args) {
        FileReader in = null;
        try {
            // 파일로부터 문자 입력 스트림 생성
            in = new FileReader("c:\\windows\\system.ini");
            int c;
            while ((c = in.read()) != -1) { // 한 문자씩 읽는다.
                System.out.print((char)c);
            }
            in.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```

파일의 끝을 만나면 read()는 -1
리턴

```
; for 16-bit app support
[386Enh]
woafont=dosapp.fon
EGA8oWOA.FON=EGA8oWOA.FON
EGA4oWOA.FON=EGA4oWOA.FON
CGA8oWOA.FON=CGA8oWOA.FON
CGA4oWOA.FON=CGA4oWOA.FON
```

```
[drivers]
wave=mmdrv.dll
timer=timer.drv
```

```
[mci]
```



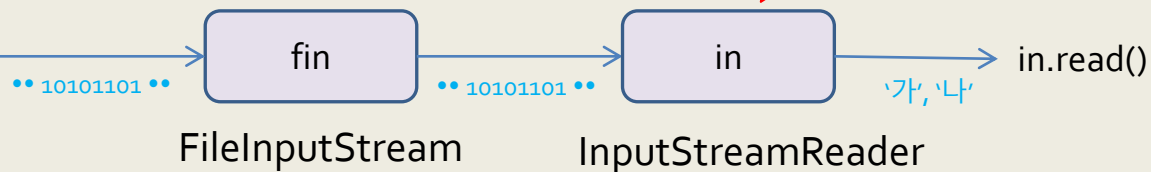
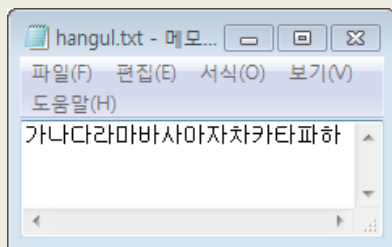
문자 집합과 InputStreamReader로 텍스트 파일 읽기

```
FileInputStream fin = new FileInputStream("c:\\tmp\\hangul.txt");  
InputStreamReader in = new InputStreamReader(fin, "MS949");
```

한글 완성형 확장형 문자 집합

```
while ((c = in.read()) != -1) {  
    System.out.print((char)c);  
}
```

문자 집합 사용
(윈도우에서 MS949)





예제 : 한글 텍스트 파일 읽기

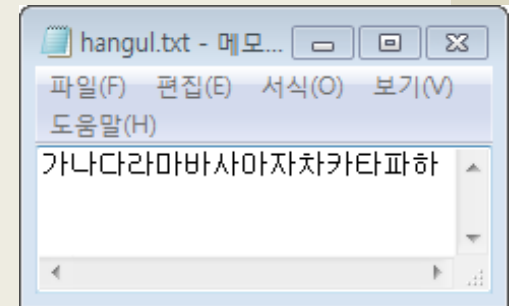
InputStreamReader를 이용하여 MS949 문자 집합으로 한글 텍스트 파일을 읽고 출력하라.

```
import java.io.*;

public class FileReadHangulSuccess {
    public static void main(String[] args) {
        InputStreamReader in = null;
        FileInputStream fin = null;
        try {
            fin = new FileInputStream("c:\\tmp\\hangul.txt");
            in = new InputStreamReader(fin, "MS949");
            int c;

            System.out.println("인코딩 문자 집합은 " + in.getEncoding());
            while ((c = in.read()) != -1) {
                System.out.print((char)c);
            }
            in.close();
            fin.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```

MS에서 만든 한글 확장
완성형 문자 집합



hangul.txt

인코딩 문자 집합은 MS949
가나다라마바사아자차카타파하



예제 : 문자 집합 지정이 잘못된 한글 텍스트 파일 읽기

InputStreamReader의 문자 집합을 US-ASCII로 지정하여 한글 파일을 출력

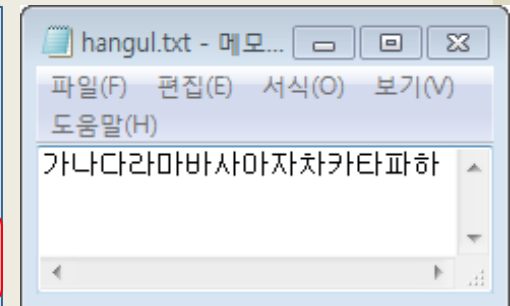
```
import java.io.*;

public class FileReadHangulFail {
    public static void main(String[] args) {
        InputStreamReader in = null;
        FileInputStream fin = null;
        try {
            fin = new FileInputStream("c:\\tmp\\hangul.txt");
            in = new InputStreamReader(fin, "US-ASCII");
            int c;

            System.out.println("인코딩 문자 집합은 " + in.getEncoding());
            while ((c = in.read()) != -1) {
                System.out.print((char)c);
            }
            in.close();
            fin.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```

문자 집합 지정이 잘못된 경우의 예를 보이기
위해 일부러 틀린 문자 집합 지정

문자 집합 지정이 잘못되어
읽은 문자가 제대로 인식되지 못함.
출력 결과가 깨짐



hangul.txt

인코딩 문자 집합은 ASCII
????????????????????????????????



FileWriter 사용 예

- ❑ c:\test.txt 파일에 문자 출력 스트림을 생성하는 코드

```
FileWriter fout = new FileWriter("c:\\tmp\\test.txt");
```

- ❑ 파일에 문자 출력

```
FileWriter fout = new FileWriter("c:\\tmp\\test.txt");  
fout.write('A'); // 문자 'A' 출력  
fout.close();
```



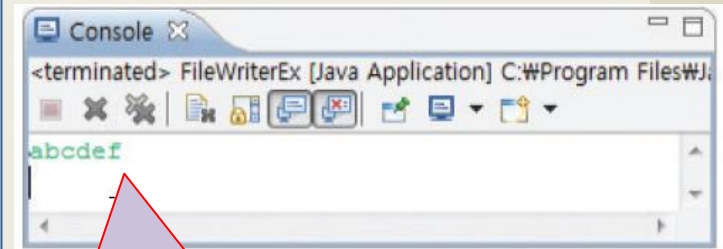
예제 : 키보드 입력을 파일로 저장하기

키보드로부터 입력받은 데이터를 c:\tmp\test.txt 파일에 저장하는 코드

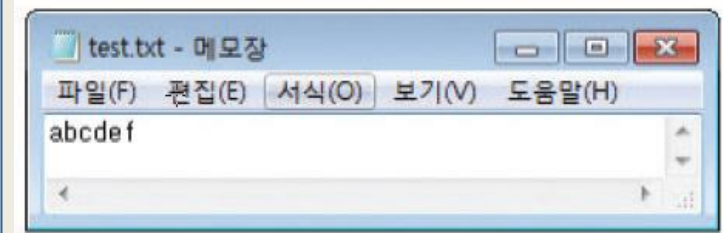
```
import java.io.*;

public class FileWriterEx {
    public static void main(String[] args) {
        InputStreamReader in = new InputStreamReader(System.in);

        FileWriter fout = null;
        int c;
        try {
            fout = new FileWriter("c:\\tmp\\test.txt");
            while ((c = in.read()) != -1) {
                fout.write(c);
            }
            in.close();
            fout.close();
        } catch (IOException e) {
            System.out.println("입출력 오류");
        }
    }
}
```



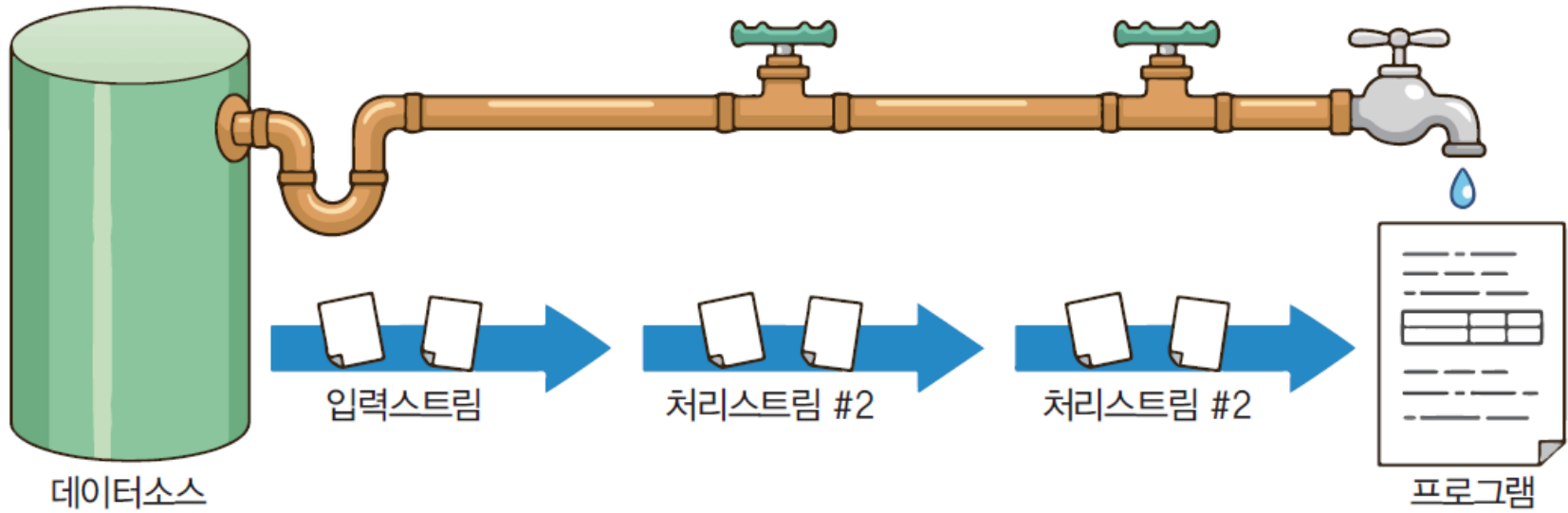
abcdef 입력 후 <Enter>
키와 ctrl-z키 입력



실행 결과 test.txt 파일 생성



스트림의 연결



- 스트림은 연결될 수 있다.

```
FileInputStream fileSt = new FileInputStream("sample.dat");  
DataInputStream dataSt = new DataInputStream(fileSt);  
int i = dataSt.readInt();
```

