

# 자바 입출력 Programming

# 프로그램 입/출력 정의

---

- **입력(Input)**
  - 데이터소스(키보드, 파일, 메모리, 네트워크 등)로부터 데이터 읽기
- **출력(Output)**
  - 데이터타겟(콘솔, 파일, 메모리, 네트워크 등)에 데이터 쓰기

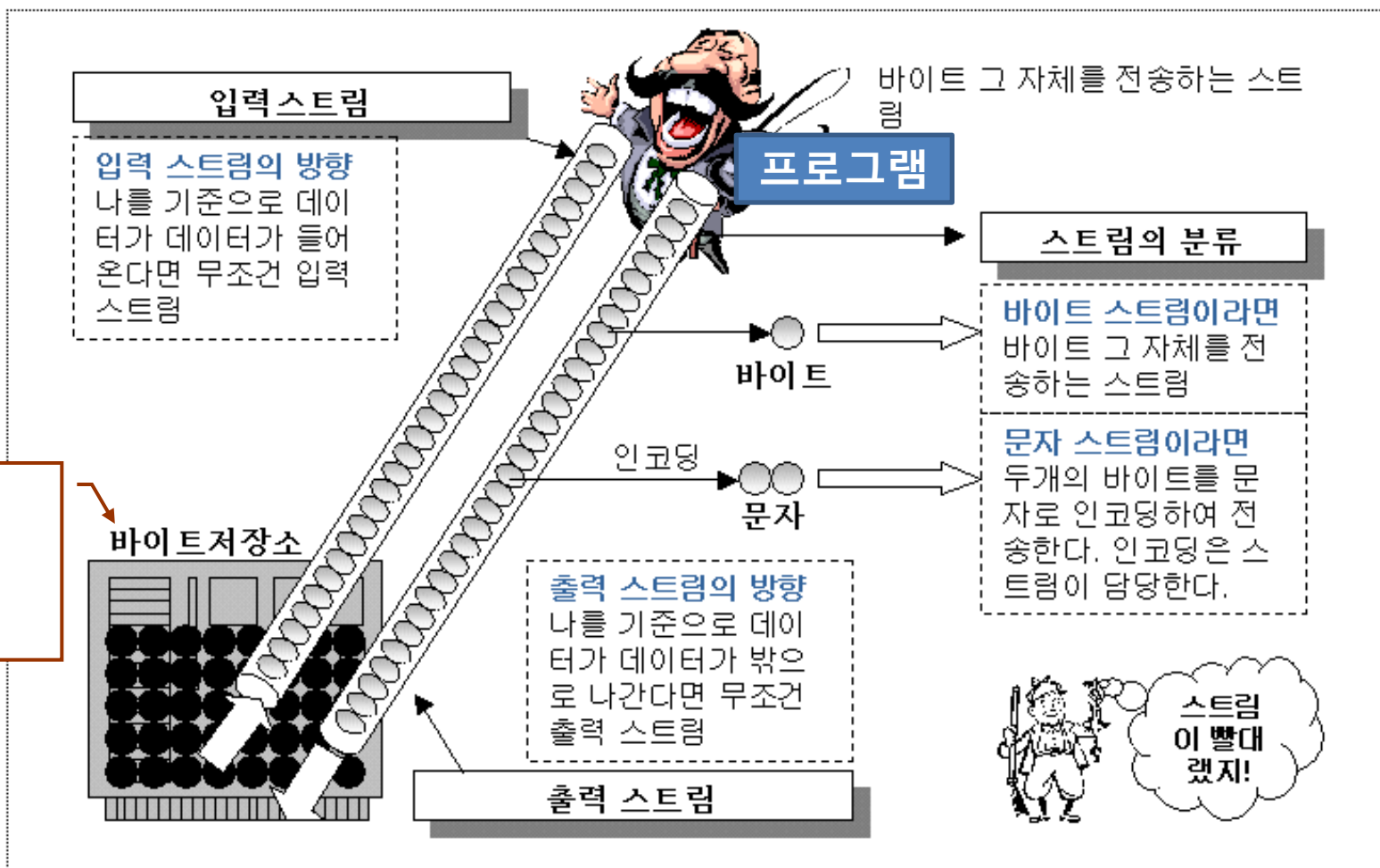
자바에서는 프로그램 입출력을 위해  
시냇물 개념을 모델링(추상화)한 **Stream API**를 제공한다

# Stream 특징

- 흐르는 물! 또는 빨대 등에 비유될 수 있다.
  - 목적지로부터 일련의 데이터(바이트)를 읽고, 쓰기 위한 중간자 역할 용도
  - 입출력 대상이나 목적지에 상관없이 스트림에서 데이터를 읽고, 쓰기만 하면 된다. - Stream API는 입출력의 일관성 제공
- 스트림은 단방향이다.
  - Stream API는 데이터 읽기 용도의 입력스트림 클래스들과 쓰기 용도의 출력스트림 클래스들을 별도로 제공한다
- 스트림은 FIFO(First In First Out) 구조이다.
  - 먼저 출력한 데이터를 먼저 읽어 들인다

“스트림은 연속적인 일련의 바이트 흐름”

# 스트림(Stream) 특징



# 스트림(Stream) 분류

---

- 데이터 목적지에 따라 여러 종류의 스트림 클래스 제공된다.
  - 입출력 프로그램을 작성하는 개발자는 데이터 목적지에 맞는 스트림 객체를 생성하고 메소드를 이용하여 읽기/쓰기를 처리 하면 된다.
- 입출력 방향에 따른 분류
  - 입력스트림(InputStream, FileInputStream 등)
  - 출력스트림(OutputStream, FileOutputStream 등)
- 입출력 데이터 종류에 따른 분류
  - 바이트(byte)스트림: 데이터를 원시의 바이트 단위 그대로 읽고 쓰기
  - 문자(character)스트림: 바이트 데이터를 문자로 가공(인코딩/디코딩)

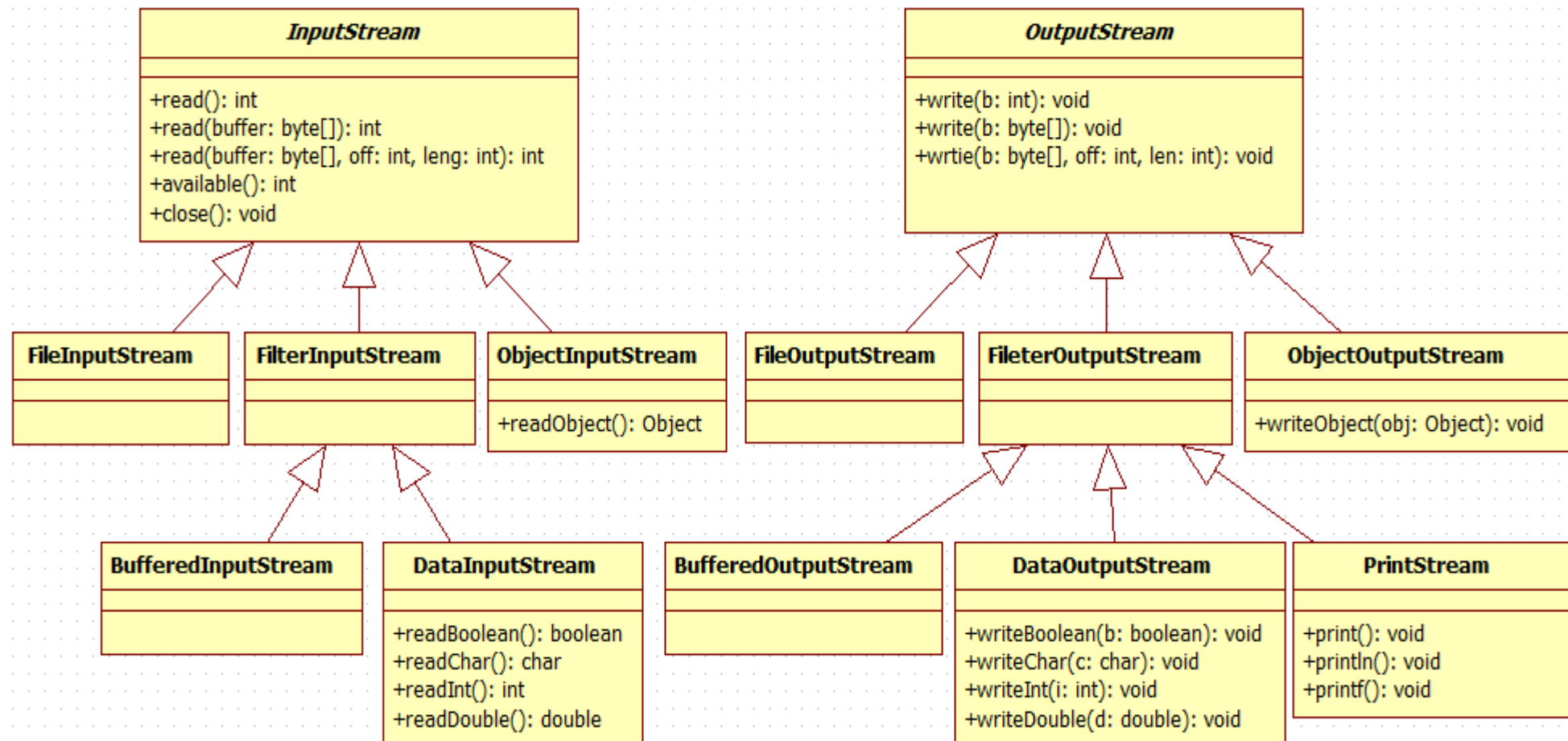
# 스트림(Stream) 구분

---

- **입출력 스트림의 용도에 따른 분류**
  - 노드 (Node)스트림: 단순한 입출력만 담당
  - 필터(Filter)스트림: 스트림의 데이터를 조작

# 스트림 APIs(java.io.\*)

- 바이트(byte) 스트림 : 1바이트 단위 입출력



# 바이트 스트림(InputStream/OutputStream)

- 데이터를 1바이트 단위로 읽고, 쓰기 위한 스트림 클래스
- 추상클래스로 존재하는 InputStream/OutputStream 이하 여러 개의 서브클래스 존재
- **입력(InputStream) 기본 메소드**
  - `int read();`
  - `int read(byte[] buffer);`
  - `int read(byte[] buffer, int offset, int length);`
- **출력(OutputStream) 기본 메소드**
  - `void write(int c);`
  - `void write(byte[] buffer);`
  - `void write(byte[] buffer, int offset, int length);`



# InputStream 클래스 상세

- InputStream 클래스는 입력스트림(바이트)의 최상위 추상 클래스
- 추상이기 때문에 new 키워드를 통해서 객체를 생성할 수 없으나 입력에 관련된 기본적인 메소드를 정의하고 있다.

메소드	기 능
abstract int read()	스트림으로부터 1바이트를 읽어 들이고, 만일 스트림의 끝일 경우 -1을 리턴한다.리턴값이 int 형이기 때문에, 읽어들인 바이트 데이터를 문자로 사용하고자 할 경우에는 char형으로 형변환하여야 한다.
int read(byte[] buffer)	스트림으로 부터 바이트 배열 사이즈 만큼 읽어서 buffer에 채운다. 리턴값은 버퍼에 채운 바이트수를 리턴한다.
int read(byte[] buffer, int offset, int length)	스트림으로 부터 length 만큼 읽어들이어서 buffer의 offset 부터 채운다
int available()	read() 메소드 호출이 블록킹 되지 않으면서 읽을 수 있는 바이트수
void close()	스트림을 닫고, 스트림과 관련된 운영체제 자원을 모두 반납.
long skip(long n)	지정한 바이트수 만큼 건너뛴다.실제로 건너 뛴 바이트 수 리턴
void mark(int readlimit)	스트림내의 현재 위치에 표시해 둔다.
void reset()	표시된 위치(marked)로 되돌린다.

# OutputStream 클래스 상세

- OutputStream 클래스는 출력스트림(바이트)의 최상위 추상 클래스
- 추상이기 때문에 new 키워드를 통해서 객체를 생성할 수 없으나 출력에 관련된 기본적인 메소드를 정의하고 있다.

메소드	기 능
void write(int b)	스트림에 한바이트를 쓴다.
void write(byte[] buffer)	스트림에 배열 buffer 만큼 쓴다.
void write(byte[] buffer, int offset, int length)	스트림에 buffer의 내용을 offset부터 length만큼 쓴다.
void flush()	스트림에 의해 쓰여진 buffer에 있던 내용을 강제로 출력한다.
void close()	스트림을 닫고, 스트림과 관련된 운영체제 자원을 모두 반납.

# InputStream 확장 클래스들

- 입력 스트림

바이트 스트림 클래스	용도
InputStream	바이트 입력 스트림을 위한 추상클래스
BufferedInputStream	바이트 버퍼 입력, 라인 해석 가능 클래스
LineNumberInputStream	바이트 입력 시, 라인 번호를 유지
ByteArrayInputStream	바이트 배열에서 읽어 들임(메모리스트림)
FileInputStream	파일에서 바이트를 읽어 들어 바이트 스트림으로 변환
FilterInputStream	필터적용(filtered) 바이트 입력을 위한 추상클래스
PushbackInputStream	읽어 들인 바이트를 되돌림(pushback)
PipedInputStream	PipedOutputStream에서 출력된 데이터를 읽어들임
StringBufferInputStream	문자열에서 읽어 들임(메모리스트림)
ObjectInputStream	직렬화 가능한 객체를 읽어 들임

# OutputStream 확장 클래스들

- 출력 스트림

바이트 스트림 클래스	용도
OutputStream	바이트 출력 스트림을 위한 추상클래스
FileOutputStream	바이트 스트림을 바이트 파일로 출력
PipedOutputStream	PipedInputStream에 출력
ByteArrayOutputStream	바이트 스트림에 바이트 배열 출력
FilterOutputStream	필터적용(filtered) 바이트 출력을 위한 추상클래스
PrintStream	Stream에 값과 객체를 프린트
BufferedOutputStream	바이트 스트림에 버퍼 출력
ObjectOutpustream	직렬화 가능한 객체를 출력

# FileInputStream 클래스 상세

- 시스템의 파일로부터 순차적인 방법으로 데이터를 한바이트씩 읽어올 수 있다.
- 지정한 파일이 존재하지 않을 경우 IOException이 발생한다.
- InputStream의 서브클래스로서 InputStream의 기본적인 메소드를 재사용하거나, 재정의하여 사용하고 있다.

생성자	설 명
FileInputStream(String name)	주어진 이름이 가르키는 파일을 바이트 스트림으로 읽기 위한 FileInputStream 객체 생성
FileInputStream(File file)	주어진 File 객체가 가르키는 파일을 읽기 위한...
FileInputStream(FileDescriptor fd)	주어진 FileDescriptor 객체가 가르키는 파일을 읽기 위한...

# File 클래스 상세

- 파일 및 디렉토리에 대한 기본 정보(크기, 변경날짜 등)를 제공하거나, 관리할 수 있 수 있는 기능만 제공할 뿐, 파일 내용에 대한 접근은 제공하지 않는다.
- 파일의 복사, 이름 변경 등의 조작을 할 경우에만 사용되며, 파일 데이터를 입출력 하기 위해서는 File 입출력 스트림이나 RandomAccessFile 로 해줘야 한다.

생성자	설 명
File (String pathName)	주어진 경로명을 가지고 새로운 File 객체를 생성한다.
File (String parent, String child)	주어진 두개의 경로명을 가지고 새로운 File 객체를 생성한다. parent는 디렉토리이며, child는 디렉토리 또는 파일일 수 있다.
File (File parent, String child)	주어진 File 객체와 문자열을 이용하여 새로운 File 객체를 생성한다. parent 객체는 디렉토리이며, child는 디렉토리 또는 파일일 수 있다.

# File 클래스 상세

- File 클래스가 제공하는 주요 메소드

메소드	설 명
String getName()	경로명이 나타내는 파일 또는 디렉토리의 이름을 얻는다. 이 이름은 경로명에 있는 마지막 이름입니다.
String getParent()	경로명의 부모 경로에 대한 경로명을 얻는다. 부모 경로명은 경로명에 있는 마지막 이름을 제외한 나머지 이름들을 포함하고 있다.
File getParentFile()	경로명의 부모 경로에 대한 File 객체를 얻는다. 부모 경로명은 경로명에 있는 마지막 이름을 제외한 나머지 이름들을 포함하고 있다.
String getPath()	경로명을 얻는다.
boolean isAbsolute()	절대 경로명인지를 얻는다. 유닉스시스템상에서는 prefix가 "/"를 포함하고 있을 경우, Win32 시스템상에서는 prefix가 "ww"와 드라이브문자 또는 "ww"를 포함하고 있을 경우 절대경로이다.
String getAbsolutePath()	절대 경로명을 얻는다.
URL toURL()	경로명을 URL로 변환하여 얻는다.

# File 클래스 상세

- File 클래스가 제공하는 주요 메소드

메소드	설 명
boolean canRead()	파일이 읽기 가능한지 여부 얻는다.
boolean canWrite()	파일이 쓰기 가능한지 여부 얻는다.
boolean exists()	파일이 존재하는지 여부 얻는다.
boolean isDirectory()	디렉토리인지 여부 얻는다.
boolean isFile()	파일인지 여부 얻는다.
boolean isHidden()	숨겨진 파일인지 여부 얻는다.
long lastModified()	마지막으로 변경된 시간을 얻는다. 시간은 그리니찌표준시간(00:00:00 GMT, January 1, 1970)을 기준으로 경과된 밀리초를 나타내며, 값이 0이면 파일이 존재하지 않거나 IOException이 발생한 경우이다.
long length()	파일의 크기를 얻는다.
boolean createNewFile()	경로명에 해당하는 파일이 존재하지 않는 경우 빈 새로운 파일 생성.



# File 클래스 상세

- File 클래스가 제공하는 주요 메소드

메소드	설 명
<code>boolean delete()</code>	경로명이 가르키는 디렉토리나 파일 삭제. 이때 비어있지 않은 디렉토리는 삭제할 수 없다.
<code>String[] list()</code>	경로명이 나타내는 디렉토리내의 파일과 디렉토리 이름에 대한 문자열 배열을 얻습니다.
<code>File[] listFiles()</code>	경로명이 나타내는 디렉토리내의 파일과 디렉토리 이름에 대한 파일 객체 배열을 얻습니다.
<code>boolean mkdir()</code>	경로명에 해당하는 디렉토리를 생성한다.
<code>boolean mkdirs()</code>	경로명에 해당하는 디렉토리를 생성 가능한지 여부를 리턴한다.
<code>boolean renameTo(File file)</code>	파일의 이름을 변경한다.
<code>static File[] listRoots()</code>	사용 가능한 파일 시스템의 루트를 얻는다.

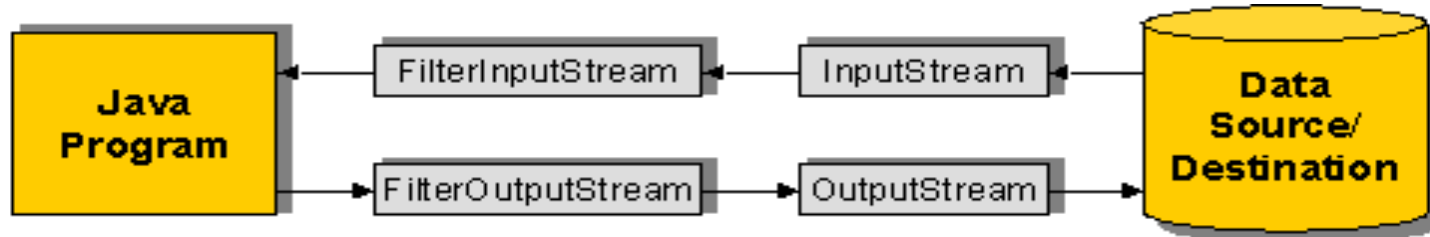
# FileOutputStream 클래스 상세

- 바이트 데이터를 파일에 저장하기 위해 사용되는 스트림 클래스이다.
- 지정한 파일이 이미 존재할 경우 그 파일에 덮어쓰게 된다.
- `OutputStream`의 서브클래스로서 `OutputStream`의 기본적인 메소드를 재사용하거나, 재정의하여 사용하고 있다.

생성자	설 명
<code>FileOutputStream(String name)</code>	주어진 이름이 가르키는 파일에 바이트 스트림으로 쓰기 위한 <code>FileOutputStream</code> 객체 생성. 기존 파일이 존재할 경우 그 내용을 지우고 새로운 파일 생성.
<code>FileOutputStream(String name, boolean append)</code>	주어진 이름이 가르키는 파일에 바이트 스트림으로 쓰기 위한 <code>FileOutputStream</code> 객체 생성. 기존 파일이 존재할 경우 주어진 <code>append</code> 값에 따라 새로운 파일을 생성하거나, 기존의 내용에 추가한다.
<code>FileOutputStream(File file)</code>	주어진 <code>File</code> 객체가 가리키는 파일에 바이트 스트림으로 쓰기 위한 <code>FileOutputStream</code> 객체 생성. 기존 파일이 존재할 경우 그 내용을 지우고 새로운 파일 생성.
<code>FileOutputStream(FileDescriptor fd)</code>	주어진 <code>FileDescriptor</code> 객체가 가리키는 파일에 바이트 스트림으로 쓰기 위한 <code>FileOutputStream</code> 객체 생성. 기존 파일이 존재할 경우 그 내용을 지우고 새로운 파일 생성

# FilterInputStream 클래스

- 일반 노드 스트림은 바이트 단위로 데이터를 읽고, 쓰는 기능만 제공
- 기존 노드 스트림에 특정 기능을 가진 필터 스트림을 연결해서 원하는 동작을 지원.(효율성, 다양한 타입으로 읽고 쓰기, 다중연결, 되돌림, 라인넘버링 등)



클래스	설 명
FilterInputStream/FilterOutputStream	필터스트림 클래스들의 최상위 클래스
LineNumberInputStream	라인 번호 입력 필터 스트림 클래스
DataInputStream/DataOutputStream	자바 기본타입으로 읽고, 쓸 수 있는 필터 스트림 클래스
BufferedInputStream/BufferedOutputStream	스트림에 버퍼를 추가한 필터 스트림 클래스
PushbackInputStream	되돌림 가능 필터 스트림 클래스
PrintStream	각각의 자료형을 문자열(텍스트)로 출력 - 디버깅때 사용

# DataStream/DataOutputStream 클래스

- 바이트 단위의 입출력 기능뿐만 아니라, 자바에서 제공하는 기본 데이터 타입별로 직접 읽고 쓸 수 있는 기능을 제공하는 필터스트림
- DataInput, DataOutput 인터페이스 구현 클래스
- 생성자 및 주요 메소드
  - API Document 참조

# BufferedInputStream/OutputStream 클래스

- 입출력의 효율성을 높이기 위해 버퍼(바이트 배열)를 내장한 필터 스트림 클래스이다.
- **BufferedInputStream**을 사용하면 읽기 동작이 있을 때 마다 목적지로부터 한바이트씩 읽는 것이 아니라, 미리 버퍼(512byte)에 담아논 데이터를 읽어들이므로 효율적이다.
- **BufferedOutputStream** 클래스는 데이터를 출력할 때, 먼저 내장한 내부 버퍼에 출력이 되고, 버퍼가 꽉 차거나, **flush()**, **close()** 메소드가 호출될 때 내부 버퍼의 내용이 출력되게 된다.

생성자/메소드	설 명
BufferedInputStream(InputStream in)	주어진 바이트 입력 스트림에 대한 BufferedInputStream 객체 생성 내부 버퍼의 크기는 디폴트 512 바이트로 설정된다.
BufferedInputStream(InputStream in, int bufferSize)	주어진 바이트 입력 스트림에 대한 BufferedInputStream 객체 생성 내부 버퍼의 크기는 주어진 bufferSize로 설정된다.
BufferedOutputStream(OutputStream out)	주어진 바이트 출력 스트림에 대한 BufferedOutputStream 객체 생성 내부 버퍼의 크기는 디폴트 512 바이트로 설정된다.
BufferedOutputStream(OutputStream out, int bufferSize)	내부 버퍼의 크기는 주어진 bufferSize로 설정된다.

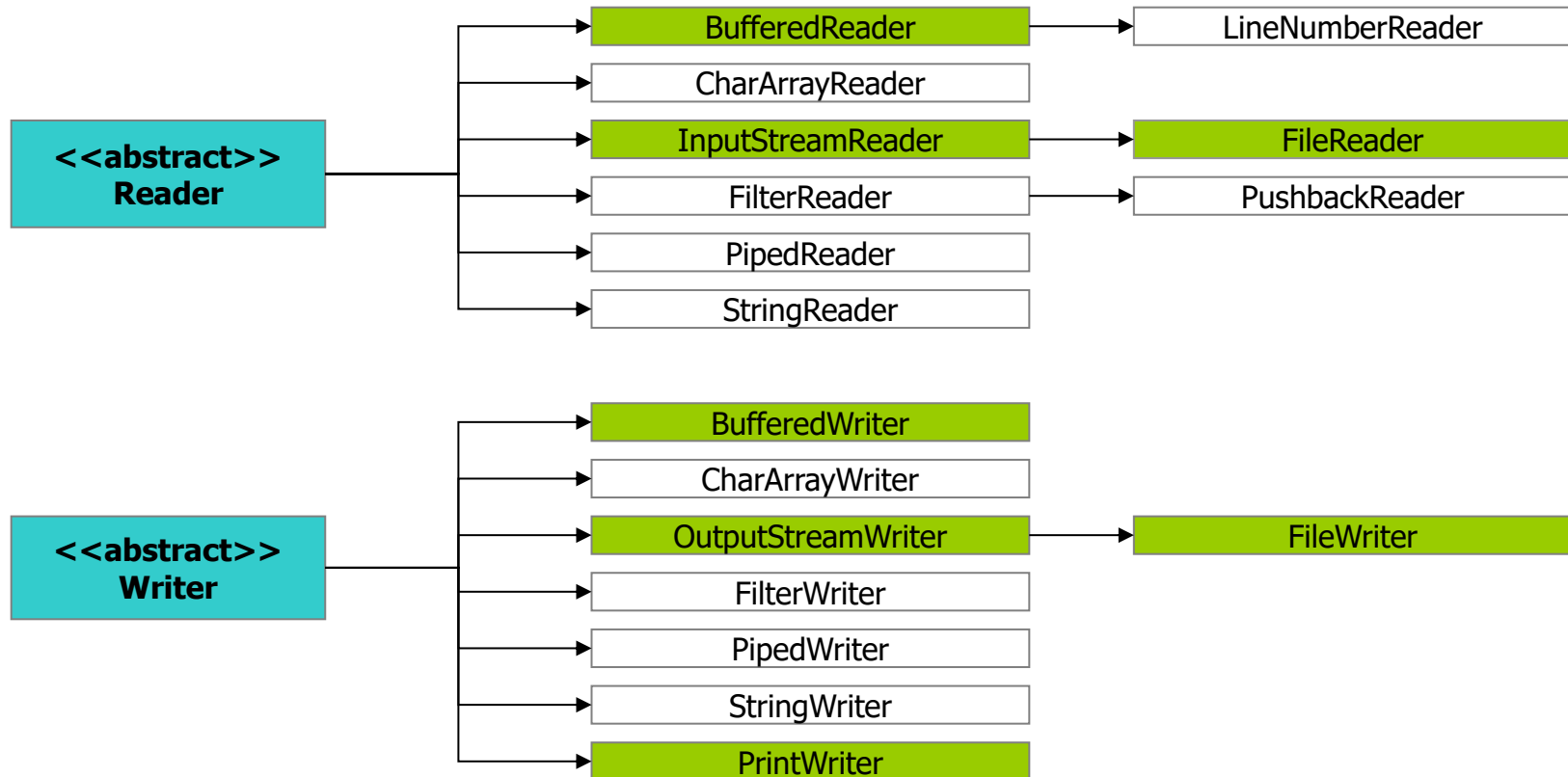
# PrintStream 클래스

---

- System.out을 통해서 도스 콘솔(표준출력)로 출력하기 위해 주로 사용된다.
  - 디버깅용으로 사용
- 기본 데이터 타입들을 OS의 문자 인코딩 방식에 맞춰 텍스트 형식으로 변환하여 출력하는 여러 종류의 메소드를 제공하고 있다.

# 문자 스트림(Reader/Writer)

- 문자(character) 스트림: 2바이트 단위 문자 인코딩/디코딩 처리



# 문자 스트림(Reader/Writer)

---

- 특별한 문자 인코딩에 상관없이 프로그램 작성 가능
  - 읽기: 바이트 입력스트림으로 부터 바이트를 읽혀들여 유니코드에 해당하는 문자로 변환
  - 쓰기: 인코딩에 따라 문자들을 바이트로 변환하고, 바이트 출력 스트림에 쓰기
- Reader/Writer(추상클래스) 이하 여러 개의 서브클래스 존재
- 입력(Reader)
  - `int read();`
  - `int read(char[] buffer);`
  - `int read(char[] buffer, int offset, int length);`
- 출력(Writer)
  - `void write(int c);`
  - `void write(char[] buffer);`
  - `void write(char[] buffer, int offset, int length);`



# Reader 확장 클래스들

- 입력 스트림

문자 스트림 클래스	용도
Reader	문자 입력 스트림을 위한 추상클래스
BufferedReader	문자 버퍼 입력, 라인 해석 가능 클래스
LineNumberReader	문자 입력 시, 라인 번호를 유지
CharArrayReader	문자 배열에서 읽어 들임
InputStreamReader	바이트 스트림을 문자 스트림으로 변환해 주는 브리지 스트림
FileReader	파일에서 바이트를 읽어 들여 문자 스트림으로 변환
FilterReader	필터 적용 문자 입력을 위한 추상클래스
PushbackReader	읽어들인 문자를 되돌림
PipedReader	PipedWriter에서 읽어 들임
StringReader	문자열에서 읽어 들임

# Writer 확장 클래스들

- 출력 스트림

문자 스트림 클래스	용도
Writer	문자 출력 스트림을 위한 추상클래스
BufferedWriter	문자 스트림에 버퍼 출력
CharArrayWriter	문자 스트림에 문자 배열 출력
FilterWriter	필터 적용 문자 출력을 위한 추상클래스
OutputStreamWriter	문자스트림을 바이트 스트림으로 변환해 주는 브리지 스트림
FileWriter	문자스트림을 바이트 파일로 변환
PrintWriter	Writer에 값과 객체를 프린트
PipedWriter	PipedReader에 출력
StringWriter	문자열 출력