KH정보교육원

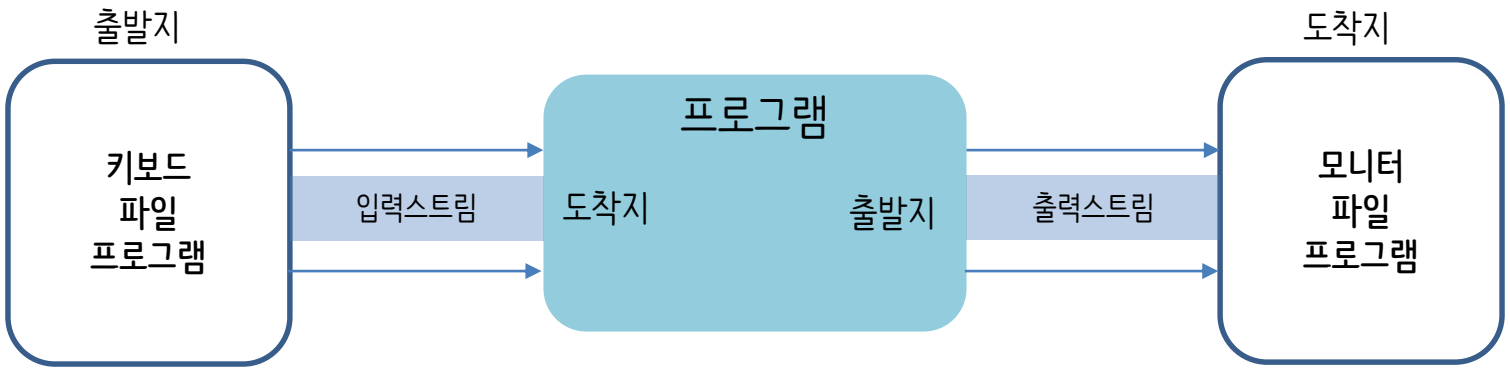
입출력(I/O)

입출력(I/O)

KH정보교육원

입출력(I/O)이란?

- Input과 Output의 약자로, 컴퓨터 내부 또는 외부 장치와 프로그램 간의 데이터를 주고 받는 것을 말한다.
- 장치와 입출력을 위해서는 하드웨어 장치에 직접 접근이 필요하고, 다양한 매체에 존재하는 데이터들을 사용하기 위해 입출력 데이터를 처리할 공통적인 방법으로 스트림을 이용한다.

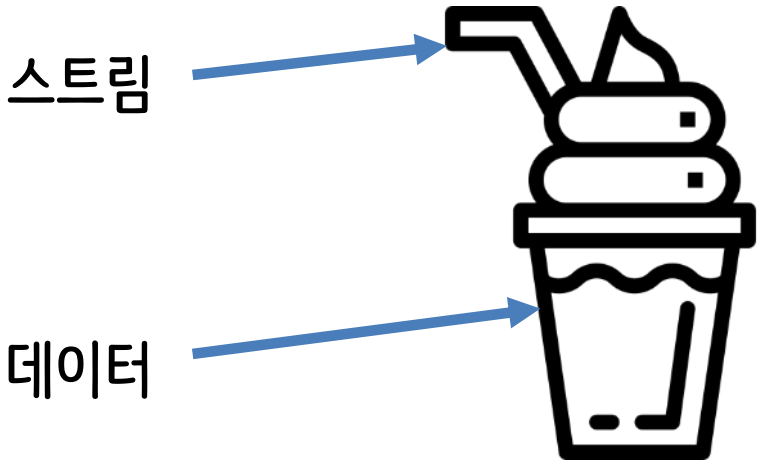


입출력(I/O)

KH정보교육원

스트림이란?

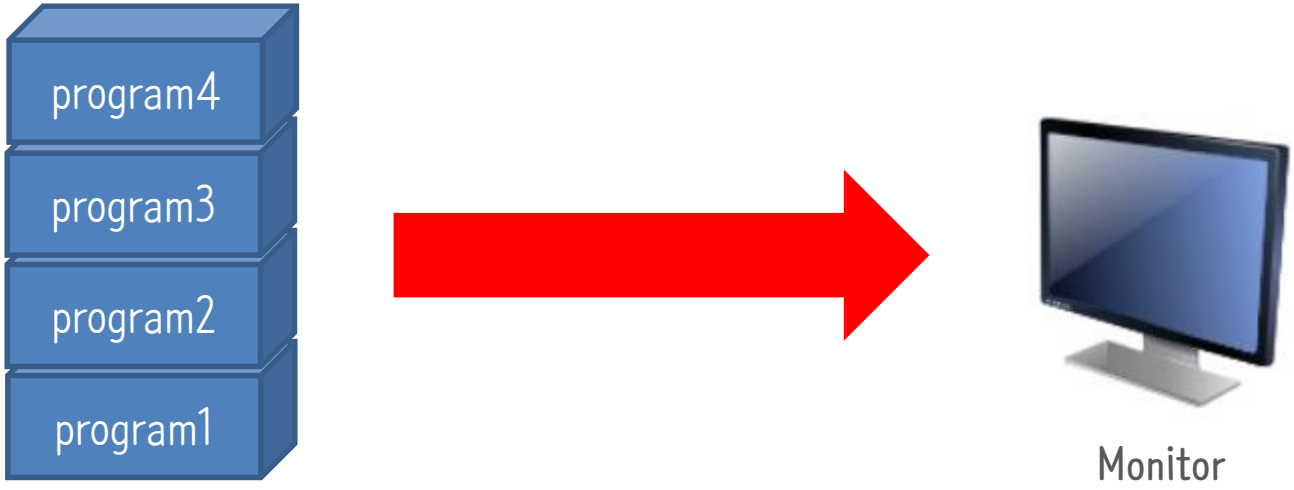
- 입출력 장치에서 데이터를 읽고 쓰기 위하여 자바에서 제공하는 class이다. 각각의 장치마다 연결할 수 있는 각각의 스트림이 존재한다.
- 단, 하나의 스트림으로 입출력을 동시에 수행할 수 없고, 입출력을 동시에 수행하려면 2개의 스트림이 필요하다. (단방향 통신)



입출력(I/O)

KH정보교육원

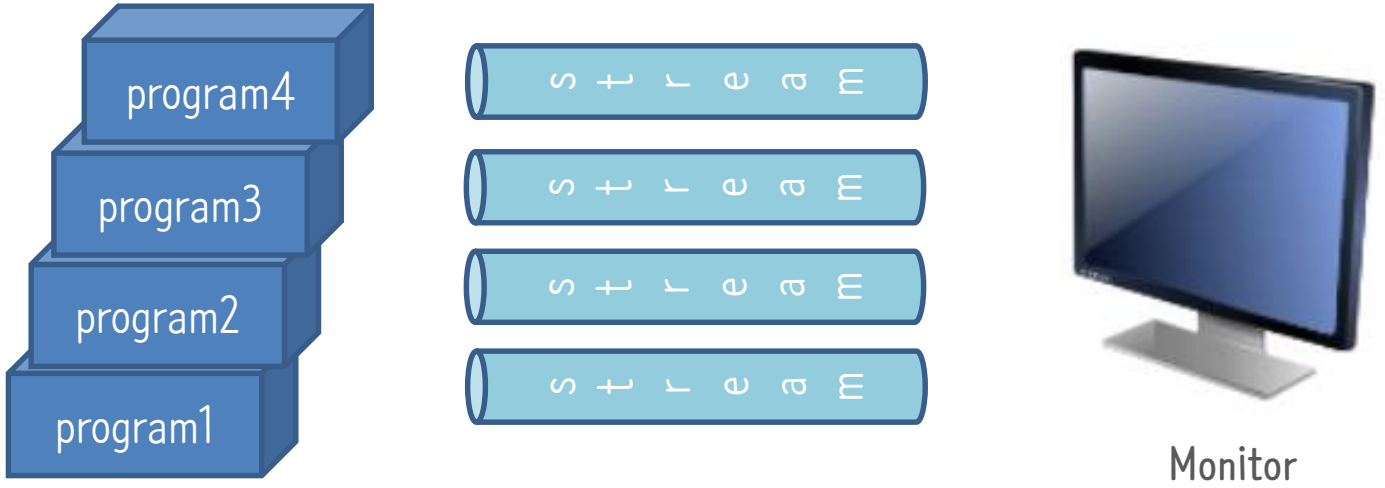
스트림은 왜 필요한가?



입출력(I/O)

KH정보교육원

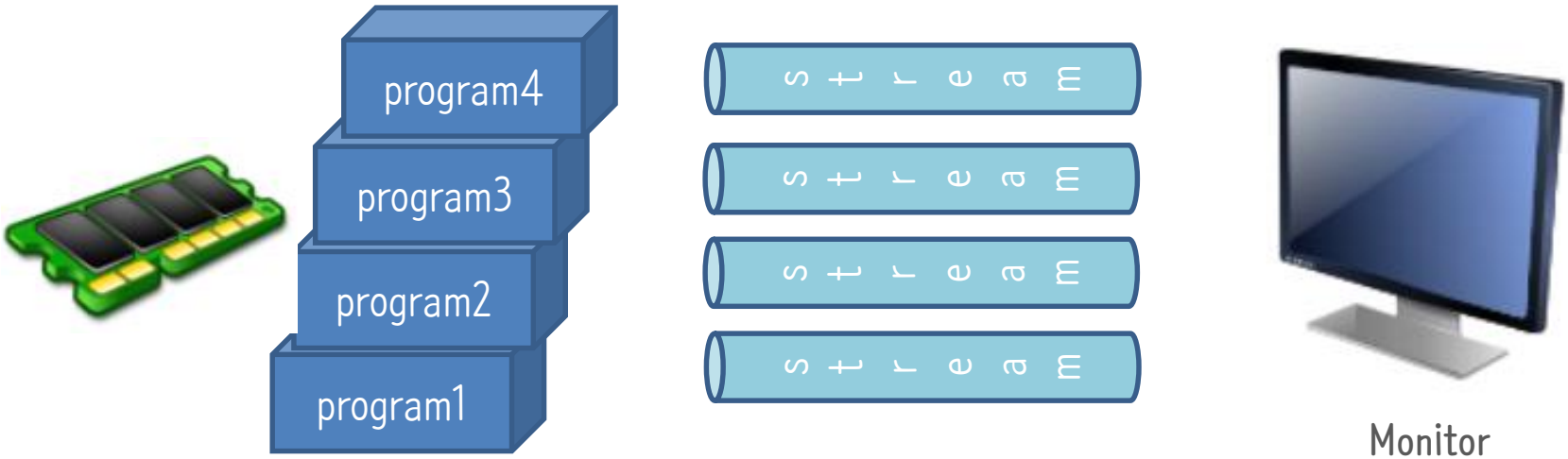
스트림은 왜 필요한가? - 출력스트림



입출력(I/O)

KH정보교육원

스트림은 왜 필요한가? - 출력스트림



입출력(I/O)

KH정보교육원

입력스트림이란?

Keyboard

입출력(I/O)

KH정보교육원

ASCII CODE란?

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

입출력(I/O)

KH정보교육원

ASCII CODE란?

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

KR — EUC-KR

JP — EUC-JP

CN — EUC-CN

입출력(I/O)

KH정보교육원

UNICODE란?

	AC0	AC1	AC2	AC3	AC4	AC5
0	가 AC00	감 AC10	갠 AC20	갰 AC30	걀 AC40	걬 AC50
1	각 AC01	갑 AC11	갡 AC21	갱 AC31	걁 AC41	걭 AC51
2	갂 AC02	값 AC12	갢 AC22	갲 AC32	걂 AC42	걮 AC52
3	갃 AC03	갓 AC13	갣 AC23	갳 AC33	걃 AC43	걯 AC53

	304	305	306	307	308	309
0		ぐ 3050	だ 3060	ば 3070	む 3080	ゐ 3090
1	ぁ 3041	け 3051	ち 3061	ぱ 3071	め 3081	ゑ 3091
2	あ 3042	げ 3052	ぢ 3062	ひ 3072	も 3082	を 3092
3	い 3043	こ 3053	っ 3063	び 3073	や 3083	ん 3093

HEX		C		J	K
4E00 — 1.0	一 G0-523B	一 HB1-A440	一 T1-4421	一 J0-306C	一 K0-6C89
4E01 — 1.1	丁 G0-3621	丁 HB1-A442	丁 T1-4423	丁 J0-437A	丁 K0-6F4B
4E02 — 1.1	丂 G5-3021		丂 T4-2126	丂 J14-2122	
4E03 — 1.1	七 G0-465F	七 HB1-A443	七 T1-4424	七 J0-3C37	七 K0-7652
4E04 — 1.1	丄 GE-3021	丄 H-9EB3	丄 T3-2126	丄 J1-3022	

입출력(I/O)

KH정보교육원

스트림의 종류 – 바이트기반 스트림

- 바이너리 데이터와 숫자를 읽고, 쓰기 위해서 사용

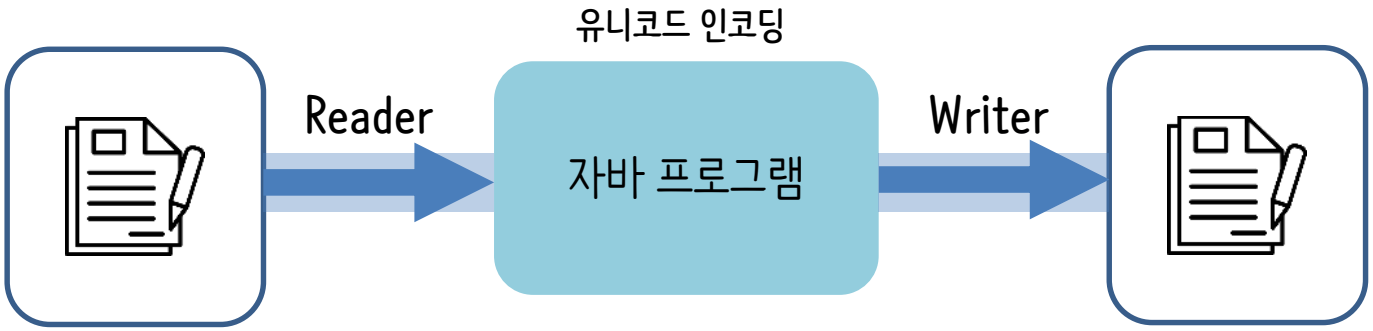
  예) 그림 파일, 동영상 파일, 응용프로그램 파일 등
- 바이트 단위로 입출력 작업

입출력(I/O)

KH정보교육원

스트림의 종류 – 문자기반 스트림

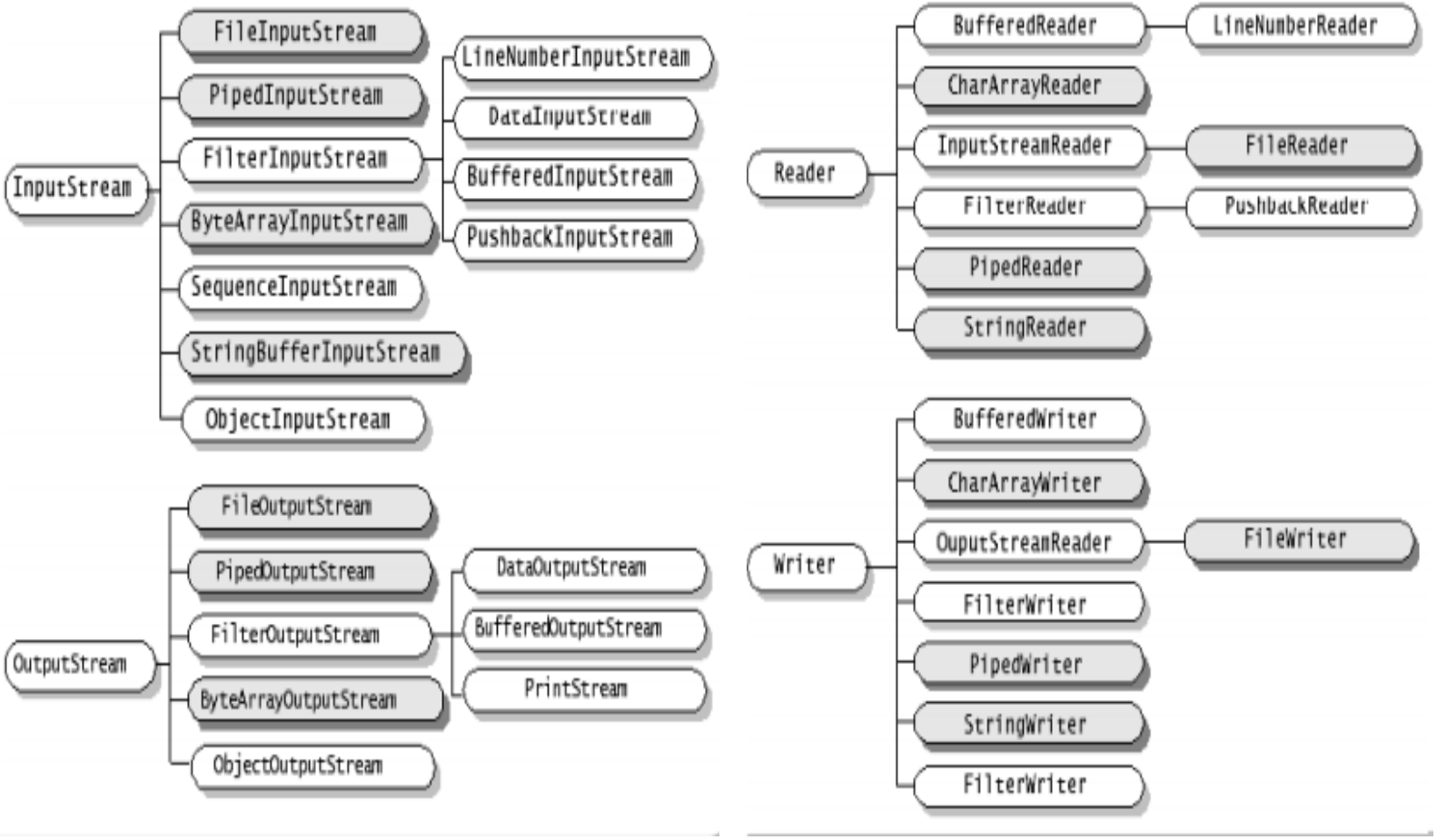
- 문자기반 스트림은 문자 데이터를 읽고 쓰기 위해서 사용
- 문자를 자동으로 인코딩



입출력(I/O)

KH정보교육원

스트림의 종류

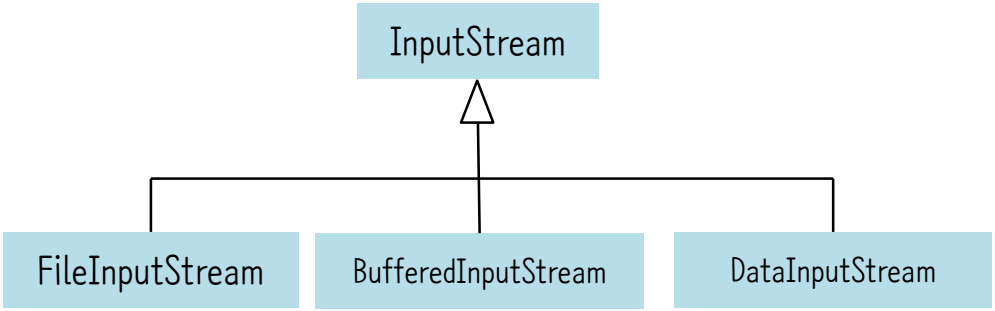


입출력(I/O)

KH정보교육원

InputStream

바이트 기반 입력 스트림의 최상위 클래스로 추상클래스이다.



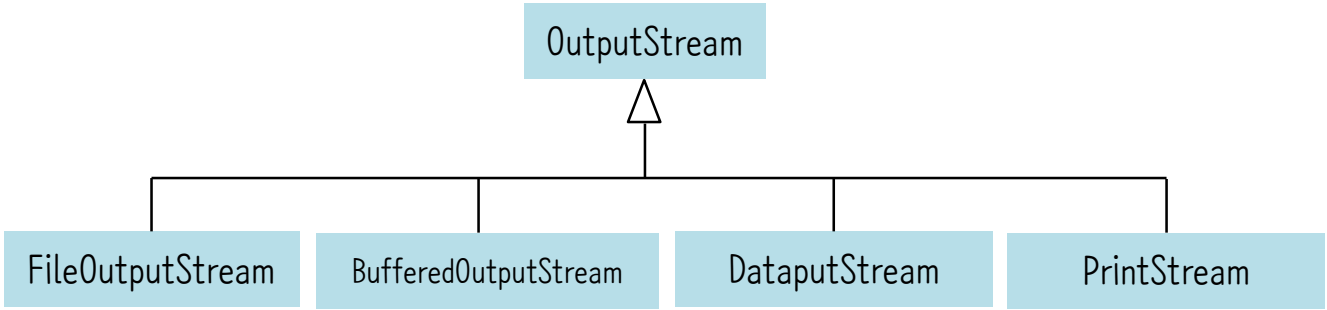
리턴타입	메소드	설명
int	read()	입력 스트림으로부터 1 바이트를 읽고, 읽은 바이트를 리턴
int	read(byte[] b)	입력 스트림으로부터 읽은 바이트들을 매개값으로 주어진 바이트배열 b에 저장하고 실제로 읽은 바이트 수를 리턴
int	read(byte[] b, int off, int len)	입력 스트림으로부터 len 개의 바이트만큼 읽고 매개값으로 주어진 바이트 배열 b[off]부터 len개 까지를 저장 그리고 실제로 읽은 바이트 수인 len 개를 리턴 만약 len개를 모두 읽지 못하면 실제로 읽은 바이트 수를 리턴
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

입출력(I/O)

KH정보교육원

OutputStream

바이트 기반 출력 스트림의 최상위 클래스로 추상클래스이다.



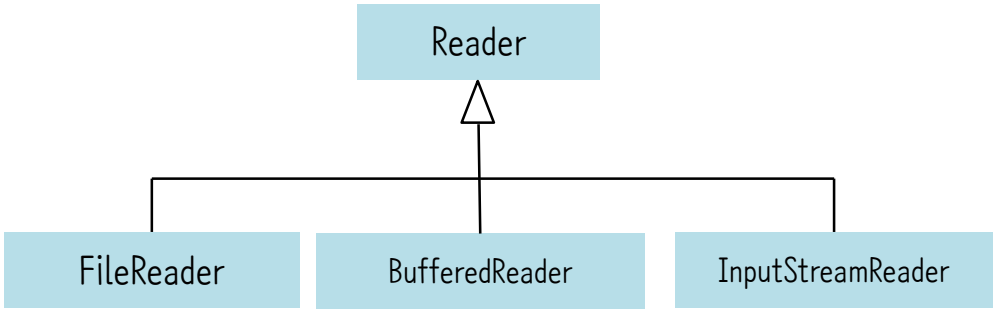
리턴타입	메소드	설명
void	write(int b)	출력 스트림으로 1바이트를 보낸다.
void	write(byte[] b)	출력 스트림에 매개값으로 주어진 바이트 배열 b의 모든 바이트를 보낸다.
void	write(byte[] b, int off, int len)	출력 스트림에 매개값으로 주어진 바이트 배열 b[off]부터 len 개 까지의 바이트를 보낸다.
void	flush()	버퍼에 잔류하는 모든 바이트를 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

입출력(I/O)

KH정보교육원

Reader

문자 기반 입력 스트림의 최상위 클래스로 추상클래스이다.



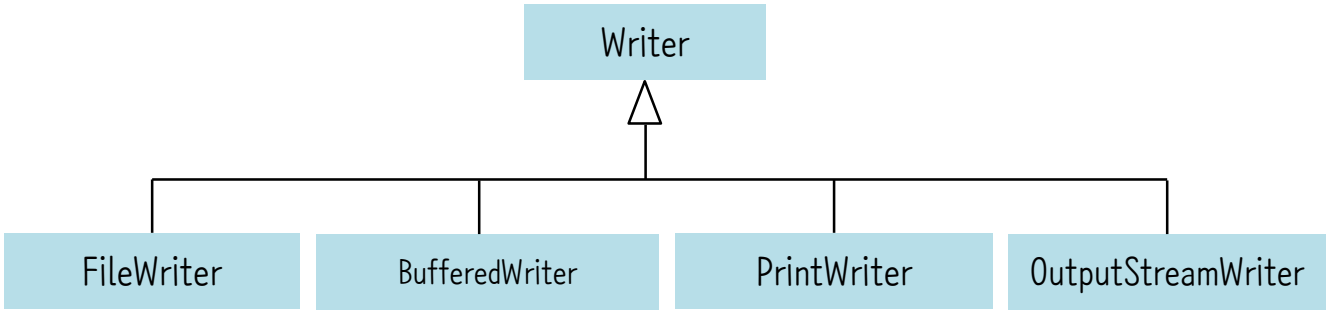
리턴타입	메소드	설명
int	read()	입력 스트림으로부터 한 개의 문자를 읽고 리턴한다.
int	read(char[] c)	입력 스트림으로부터 읽은 문자들을 매개값으로 주어진 문자 배열 c에 저장하고 실제로 읽은 문자 수를 리턴
int	read(char[] c. int off, int len)	입력 스트림으로부터 len개의 문자를 읽고 매개값으로 주어진 문자 배열 c[off]부터 len개까지 저장한다. 그리고 실제로 읽은 문자 수인 len개를 리턴
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

입출력(I/O)

KH정보교육원

Writer

문자 기반 출력 스트림의 최상위 클래스로 추상클래스이다.



리턴타입	메소드	설명
void	write(int c)	출력 스트림으로 매개값이 주어진 한 문자를 보낸다.
void	write(char[] c)	출력 스트림에 매개값으로 주어진 문자 배열 c의 모든 문자를 보낸다.
void	write(char[] c, int off, int len)	출력 스트림에 매개값으로 주어진 문자 배열 c[off]부터 len개 까지의 문자를 보낸다
void	write(String str)	출력 스트림에 매개값으로 주어진 문자열을 전부 보낸다.
void	write(String str, int off, int len)	출력 스트림에 매개값으로 주어진 문자열 off순번부터 len개까지의 문자를 보낸다.
void	flush()	버퍼에 잔류하는 모든 문자열을 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

입출력(I/O)

KH정보교육원

File 클래스

파일시스템의 파일을 표현하는 클래스로 파일크기, 파일속성, 파일이름 등의 정보를 제공하며 파일 생성 및 삭제 기능도 제공한다.

File 객체 생성

```
File file = new File("c:₩₩temp₩₩file.txt");
```

파일 및 디렉토리 생성 및 삭제 메소드

리턴타입	메소드	설명
boolean	createNewFile()	새로운 파일을 생성
boolean	mkdir()	새로운 디렉토리를 생성
boolean	mkdirs()	경로상에 없는 모든 디렉토리를 생성
boolean	delete()	파일 또는 디렉토리 삭제

입출력(I/O)

KH정보교육원

파일 및 디렉토리의 정보를 리턴하는 메소드

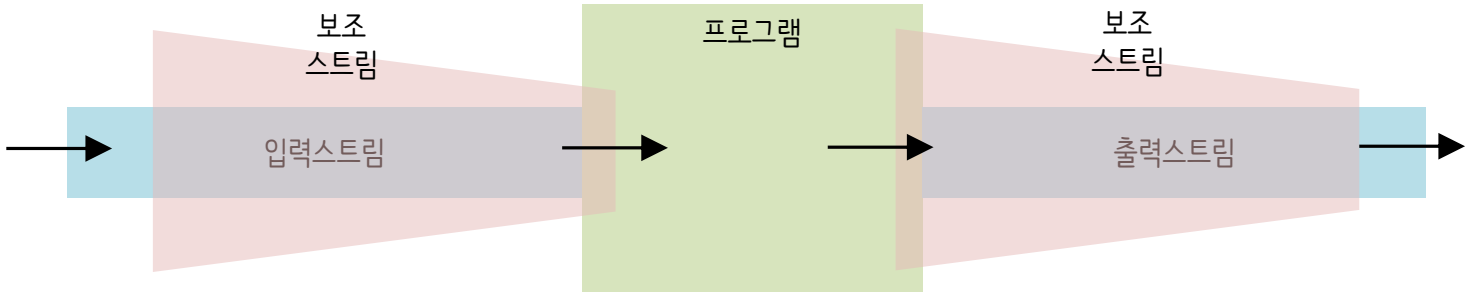
리턴타입	메소드	설명
boolean	canExcute()	실행할 수 있는 파일인지 여부
boolean	canRead()	읽을 수 있는 파일인지 여부
boolean	canWrite()	수정 및 저장할 수 있는 파일인지 여부
String	getName()	파일의 이름을 리턴
String	getParent()	부모 디렉토리를 리턴
File	getParentFile()	부모 디렉토리를 File 객체로 생성 후 리턴
String	getPath()	전체 경로를 리턴
boolean	isDirectory()	디렉토리인지 여부
boolean	isFile()	파일인지 여부
boolean	isHidden()	숨김 파일인지 여부
long	lastModified()	마지막 수정 날짜 및 시간을 리턴
long	length()	파일의 크기 리턴
String[]	list()	디렉토리에 포함된 파일 및 서브디렉토리 목록 중에 FilenameFilter에 맞는 것만 String배열로 리턴
String[]	list(FilenameFilter filter)	디렉토리에 포함된 파일 및 서브디렉토리 목록 중에 FilenameFilter에 맞는 것만 String 배열로 리턴
File[]	listFiles()	디렉토리에 포함된 파일 및 서브 디렉토리 목록 전부를 File 배열로 리턴
File[]	listFile(FilenameFilter filter)	디렉토리에 포함된 파일 및 서브디렉토리 목록 중에 FilenameFilter에 맞는 것만 File 배열로 리턴

입출력(I/O)

KH정보교육원

보조스트림(프로세스 스트림)

스트림의 기능을 향상시키거나 새로운 기능을 추가하기 위해 사용된다. 보조스트림은 실제 데이터를 주고 받는 스트림이 아니기 때문에 입출력 처리를 할 수 없고, 스트림을 먼저 생성한 다음 이를 이용해서 보조스트림을 생성해야 한다.



입출력(I/O)

KH정보교육원

보조스트림(프로세스 스트림)의 종류

문자 변환(Input/OutputStreamReader), 입출력 성능 향상(BufferedInput/OutputStream), 기본 데이터 타입 출력(DataInput/OutputStream), 객체 입출력(ObjectInput/OutputStream) 등의 기능을 제공하는 보조스트림이 있다.

사용 예시)

```
//기반 스트림 생성
FileInputStream fis = new FileInputStream("sample.txt");
//보조스트림 생성
BufferedInputStream bis = new BufferedInputStream(fis);
//보조스트림으로부터 데이터 읽어옴
bis.read();
```

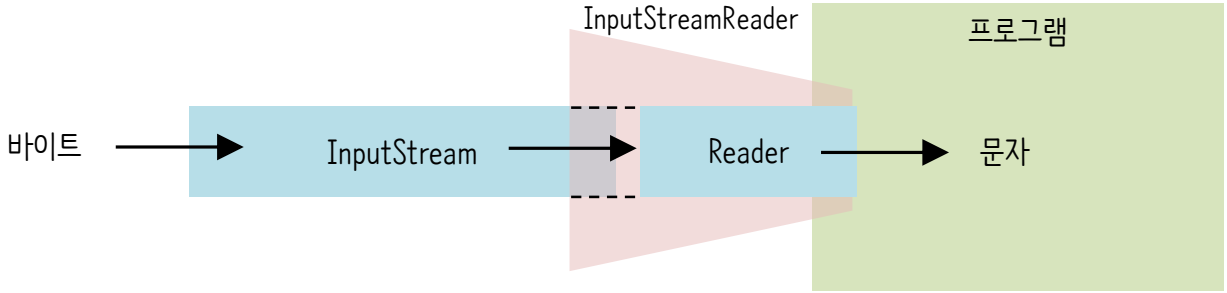
입출력(I/O)

KH정보교육원

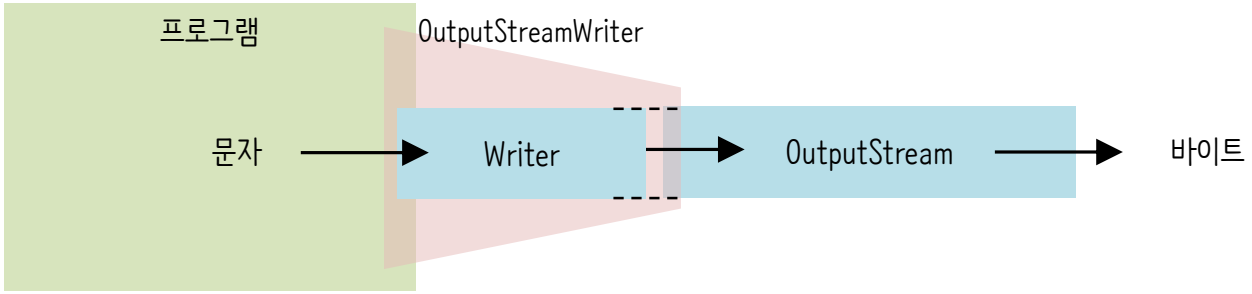
문자 변환 보조스트림

소스 스트림이 바이트 기반 스트림이지만 데이터가 문자일 경우 사용한다. Reader와 Writer는 문자 단위로 입출력을 하기 때문에 데이터가 문자인 경우에 바이트 기반 스트림보다 편리하게 사용할 수 있다.

InputStreamReader



OutputStreamWriter



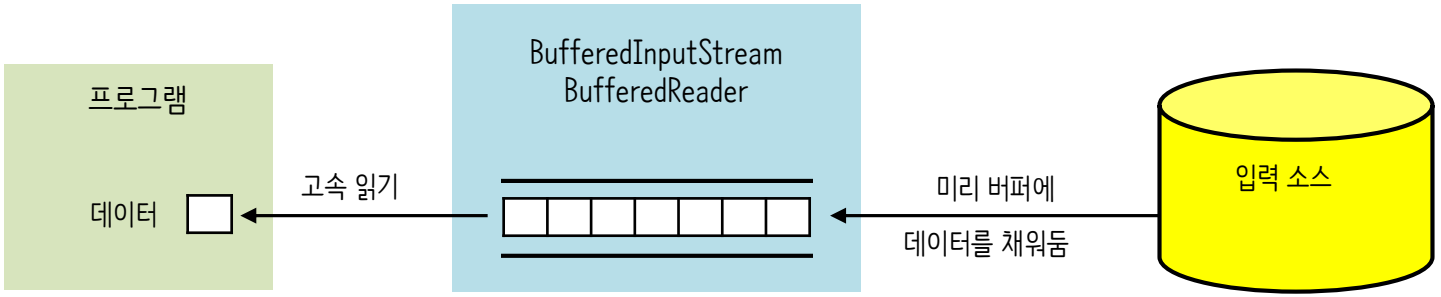
KH정보교육원

입출력 성능 향상 보조스트림

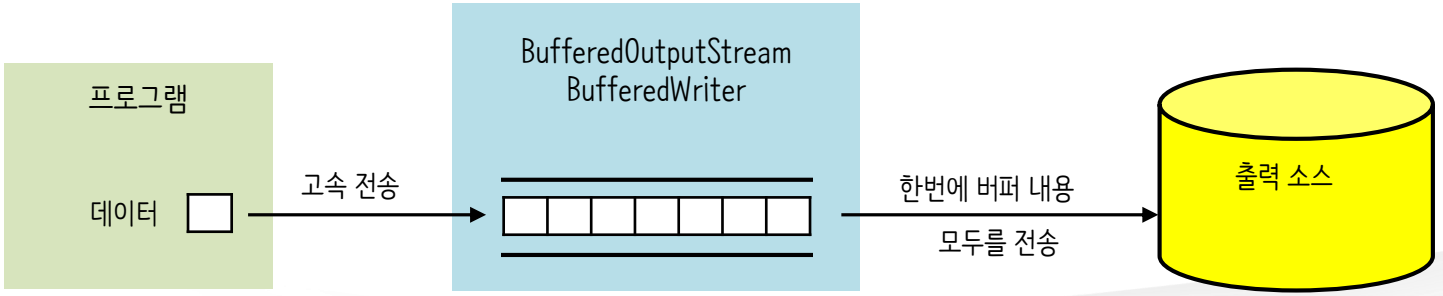
느린 속도로 인해 입출력 성능에 영향을 미치는 입출력 소스를 이용하는 경우(하드디스크, 느린 네트워크)사용한다.

입출력 소스와 직접 작업하지 않고 버퍼(buffer)에 데이터를 모아 한꺼번에 작업을 하여 실행 성능이 향상된다.(입출력 횟수 줄임)

BufferedInputStream



BufferedOutputStream



입출력(I/O)

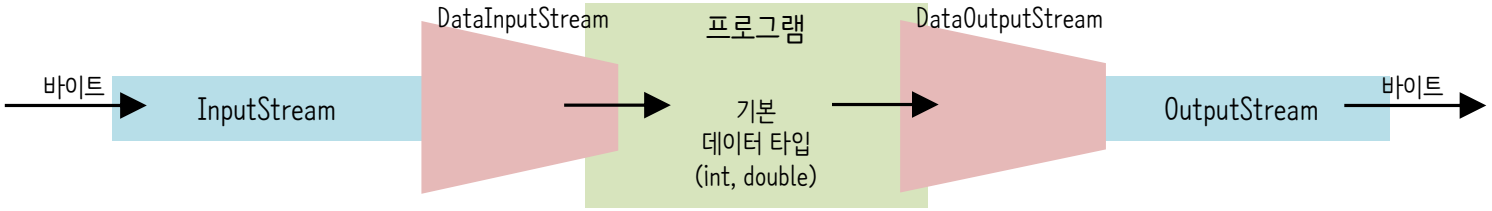
KH정보교육원

기본 데이터 타입 출력 보조스트림

기본 자료형별 데이터 읽고 쓰기가 가능하도록 기능을 제공한다.

단, 입력된 자료형의 순서와 출력될 자료형의 순서가 일치해야 한다.

DataInputStream/DataOutputStream



DataInputStream		DataOutputStream	
boolean	readBoolean()	void	writeBoolean(Boolean v)
byte	readByte()	void	writeByte(int v)
char	readChar()	void	writeChar(int v)
double	readDouble()	void	writeDouble(double v)
float	readFloat()	void	writeFloat(float v)
int	readInt()	void	writerInt(int v)
long	readLong()	void	writeLong(long v)
short	readShort()	void	writeShort(int v)
String	readUTF()	void	writeUTF()

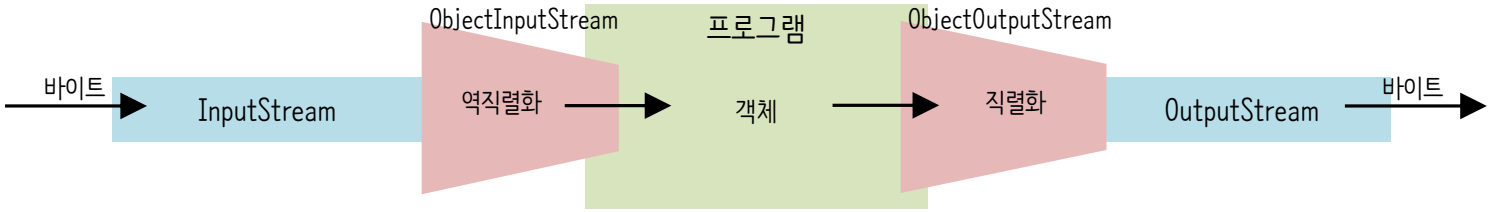
KH정보교육원

객체 입출력 보조스트림

객체를 파일 또는 네트워크로 입출력 할 수 있는 기능을 제공한다.

단, 객체는 문자가 아니므로 바이트 기반 스트림으로 데이터를 변경해 주는 작업인 직렬화를 반드시 해야 한다.

ObjectInputStream/ObjectOutputStream



입출력(I/O)

KH정보교육원

직렬화(Serializable)

Serializable 인터페이스를 implements하여 구현한다.

단,객체 직렬화 시 private 필드를 포함한 모든 필드를 바이트로 변환하지만 transient 키워드를 사용한 필드는 직렬화에서 제외한다.

역직렬화

직렬화된 객체를 역직렬화 할 때는 직렬화 했을 때와 같은 클래스를 사용한다. 하지만 클래스의 이름이 같더라도 클래스의 내용이 변경된 경우 역직렬화에 실패하게 된다.

입출력(I/O)

KH정보교육원

serialVersionUID 필드

직렬화 한 클래스와 같은 클래스임을 알려주는 식별자 역할을 한다.

컴파일시 JVM이 자동으로 serialVirsionUID 정적 필드를 추가(컴파일 할 때마다 변경됨)해 주어 별도로 작성을 하지 않아도 오류는 나지 않는다.

하지만 자동 생성시 역직렬화 과정에서 예상하지 못한 InvalidClassException을 유발할 수 있기 때문에 명시를 권장한다.

```
public class userInfo implements Serializable
        {
        private static final long serialVersionUID = 392839429392L;
        private String name;
        private int age;
        private String addr;
        }
```

KH정보교육원

transient 키워드

객체 직렬화 처리시 제외할 필드 앞에 붙여줌

```
public class userInfo implements Serializable
        {
        private static final long serialVersionUID = 392839429392L;
        private String name;
        private int age;
        private transient String addr; //직렬화에서 제외됨
        }
```