

# 15장 입출력 프로그래밍



# 입출력 프로그래밍

## ◆ File

시스템에 있는 파일이나 디렉토리를 추상화한 클래스

파일의 생성, 삭제, 크기, 읽기 쓰기 모드 등과 같이 파일 자체를 관리하는 클래스

[표 14-1] File 클래스의 주요 생성자

생성자	설명
File(String pathname)	문자열 pathname을 가지고 경로를 생성하여 File 객체를 생성한다.
File(String parent, String child)	Parent와 child 문자열을 연결한 문자열로 경로를 생성하여 File 객체를 생성한다.
File(File parent, String child)	Parent의 파일 객체와 child 문자열로 경로를 생성하여 File 객체를 생성한다.

# 입출력 프로그래밍

## ◆ File 클래스의 주요 메소드

[표 14-2] File 클래스의 주요 메서드

반환형	메서드	설명
boolean	canRead( )	파일을 읽을 수 있으면 true, 그렇지 않으면 false다.
	canWrite( )	파일을 쓸 수 있으면 true, 그렇지 않으면 false다.
	createNewFile( )	파일을 새로 생성하면 true, 그렇지 않으면 false다.
	delete( )	파일을 지우면 true, 그렇지 않으면 false다.
	exists( )	파일이나 디렉토리가 존재하면 true, 그렇지 않으면 false다.
String	getAbsolutePath( )	파일의 절대 경로를 반환한다.
	getCanonicalPath( )	파일의 정규 경로를 반환한다.
	getName( )	파일명을 반환한다.
boolean	isDirectory( )	디렉토리면 true, 그렇지 않으면 false다.
	isFile( )	파일이면 true, 그렇지 않으면 false다.
long	lastModified( )	1970년 1월 1일부터 현재까지의 시간을 밀리세컨드 초로 반환한다.
	length( )	파일의 크기를 바이트로 반환한다.
String[ ]	list( )	특정 디렉토리의 모든 파일과 자식 디렉토리를 스트링 배열로 반환한다.
boolean	mkdir( )	디렉토리를 생성하면 true, 디렉토리가 있어서 생성하지 못하면 false다.
	renameTo(File dest)	dest 파일 객체로 이름을 바꾸면 true, 그렇지 않으면 false다.

## 입출력 프로그래밍

### ◆ 입출력

자바에서는 입출력 시, 데이터를 읽고 쓸 때 스트림(Stream) 형태로 읽고 쓴다.

### ◆ 스트림(Stream)

일차원적인 데이터의 흐름으로, 데이터를 목적지로 입출력하기 위한 방법이다.

스트림에 데이터를 쓸 수 있고, 스트림에서 데이터를 읽을 수 있다.

출력 스트림(output stream) : 스트림에 데이터를 쓰는 경우

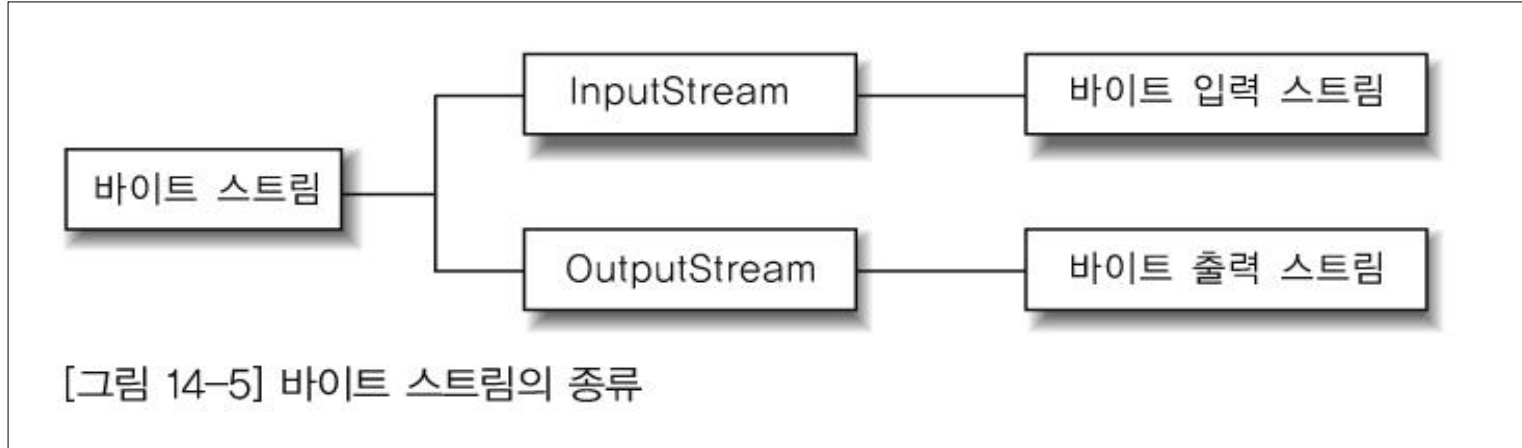
입력 스트림(input stream) : 스트림에서 데이터를 읽는 경우

# 입출력 프로그래밍

## ◆ 바이트 스트림(Stream)

1 byte를 입출력할 수 있는 스트림

일반적으로 바이트로 구성된 파일 (동영상, 이미지, 음악 파일 등)을 처리하기에 적합한 스트림이다.



## 입출력 프로그래밍

### ◆ InputStream

바이트 입력을 수행하는데 필요한 메소드를 정의하는 추상 클래스

자바 프로그램은 객체를 생성하고 생성된 객체와 바이트 스트림과 연결함으로써 파일을 연다.

자바는 다른 장치들과도 바이트 스트림을 연결할 수 있고, 프로그램이 시작되면 장치들과 연결된 세 개의 객체(System.in, System.out, System.err)를 생성한다.

System.in 객체는 키보드로 바이트를 입력할 수 있는 InputStream 객체이다.

# 입출력 프로그래밍

## ◆ InputStream 주요 메소드

[표 14-4] InputStream의 주요 메서드

반환형	메서드	설명
abstract int	read( )	스트림 데이터 1바이트를 읽어온다. 반환값은 0~255의 아스키코드값이기 때문에 문자로 나타내려면 char로 캐스팅해야 한다. 더 이상 읽을 수 없을 때는 -1을 반환한다.
int	read(byte b[ ])	스트림 데이터 1바이트를 읽어 바이트 배열에 저장하고, 읽은 수만큼 반환한다.
	read(byte b[ ], int start, int length)	스트림 데이터를 length만큼 읽어 바이트 배열 b의 start 위치에 저장하고, 읽을 수만큼 반환한다.
	available( )	읽을 수 있는 바이트 수를 반환한다.
long	skip(long n)	읽을 수 있는 바이트에서 n만큼 건너뛴다.
void	close( )	입력 스트림을 닫는다.

# 입출력 프로그래밍

## ◆ OutputStream

바이트 출력을 수행하는데 필요한 메소드를 정의하는 추상 클래스

프로그램이 시작되면 출력 스트림은 객체(System.out, System.err)를 생성한다.

[표 14-8] OutputStream의 주요 메서드

반환형	메서드	설명
abstract void	write(int b )	출력 스트림으로 b의 값을 바이트로 변환하여 쓰기한다.
void	write(byte[ ] b)	출력 스트림으로 바이트 배열 b를 쓰기한다.
	write(byte[ ] b, int start, int length)	출력 스트림으로 바이트 배열 b를 start부터 length만큼 쓰기한다.
	flush( )	출력 스트림을 통하여 쓰기를 할 때 일반적으로 버퍼에 가득차게 되면 한꺼번에 보내게 되는데, 이 메서드를 사용하게 되면 버퍼에 가득 차 있지 않더라도 버퍼의 내용을 바로 보내게 된다.
	close( )	모든 자원을 반납한다.



## 입출력 프로그래밍

### ◆ 문자 스트림 (Reader & Writer)

Reader는 입력용 문자 스트림이며, Writer는 출력용 문자 스트림이다.

바이트 스트림과는 달리 2바이트를 입출력할 수 있는 문자 기반 스트림이다.

바이트 스트림은 1바이트를 입출력하기 때문에 일반적으로 영문자로 구성된 파일 및 동영상 파일, 음악 파일의 입출력에 적합한 스트림이다.

문자 스트림은 2바이트를 입출력하기 때문에 세계 모든 언어로 구성된 파일을 입출력하기에 적합하다.

# 입출력 프로그래밍

## ◆ Reader 주요 메소드

[표 14-13] Reader의 주요 메서드

반환형	메서드	설명
int	read( )	문자 입력 스트림에서 한 개의 문자를 읽어온다. 반환값은 0에서 65535 (0x0000-0xffff) 범위의 유니코드값을 숫자로 반환한다. 문자로 나타내기 위해서는 char로 캐스팅해준다. 더 이상 읽을 수 없을 때는 -1을 반환한다.
	read(char[ ] cbuf)	문자 입력 스트림에서 문자를 하나씩 읽어 char[ ]에 저장하고, 읽은 수만큼 반환한다.
abstract int	read(char[ ] b, int start, int length)	문자 입력 스트림에서 문자를 length만큼 읽어 char[ ]의 start 위치에 저장하고, 읽을 수만큼 반환한다.
int	skip(long n)	문자 입력 스트림에서 n만큼의 문자를 건너뛴다.
abstract void	close( )	문자 입력 스트림을 닫는다.

2 byte를  
읽을 수 있다

# 입출력 프로그래밍

## ◆ Writer 주요 메소드

[표 14-16] Writer의 주요 메서드

반환형	메서드	설명
abstract void	write(int c )	문자 출력 스트림으로 c값을 char로 변환하여 쓰기한다.
void	write(char[ ] cbuf)	문자 출력 스트림으로 문자 배열 b를 쓰기한다.
abstract void	write(char[ ] cbuf, int start, int length)	문자 출력 스트림으로 문자 배열 b를 start부터 length만큼 쓰기한다.
void	write(String str)	문자 출력 스트림으로 문자열 str를 쓰기한다.
abstract void	flush( )	버퍼가 가득 차 있지 않더라도 버퍼의 내용을 바로 보내게 된다.
abstract void	close( )	문자 출력 스트림을 닫는다.

2 byte를  
출력한다

# 입출력 프로그래밍

## ◆ FileInputStream

시스템에 있는 모든 파일을 읽을 수 있는 기능을 제공한다.

파일을 읽을 때는 파일의 경로와 파일 객체를 생성자의 매개 변수로 설정할 수 있다.

만약 파일이 존재하지 않으면 FileNotFoundException이 발생하게 된다.

[표 14-5] FileInputStream의 주요 생성자

생성자	설명
FileInputStream(String name)	name이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, 이것을 매개변수로 FileInputStream 객체를 생성한다.
FileInputStream(File file)	File 객체를 이용하여 FileInputStream 객체를 생성한다.

# 입출력 프로그래밍

## ◆ FileOutputStream

시스템에 있는 모든 파일에 쓸 수 있는 기능을 제공한다.

해당 경로에 해당 파일이 있는 경우에는 파일을 생성하지 않으며, 없는 경우에 생성하게 된다.

[표 14-9] FileOutputStream의 주요 생성자

생성자	설명
FileOutputStream (String name)	name이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, 이것을 매개변수로 FileOutputStream 객체를 생성한다.
FileOutputStream (String name, boolean append)	name이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, append가 true이면 이어쓰기의 기능을 하고, false이면 덮어쓰기를 한다. 이 두 개의 매개변수로 FileOutputStream 객체를 생성한다.
FileOutputStream(File file)	File 객체를 이용하여 FileOutputStream 객체를 생성한다.
FileOutputStream (File file, boolean append)	이 두 개의 매개변수로 FileOutputStream 객체를 생성한다. append의 의미는 두 번째 생성자와 동일하다.

# 입출력 프로그래밍

## ◆ FileReader 와 FileWriter

문자 스트림으로 한 문자를 읽기 때문에 화면에 출력하더라도 한글이 깨지는 현상이 일어나지 않는다.

[표 14-14] FileReader의 주요 생성자

생성자	설명
FileReader(String filename)	filename이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, 이것을 매개변수로 FileReader 객체를 생성한다.
FileReader(File file)	File 객체를 이용하여 FileReader 객체를 생성한다.

[표 14-17] FileWriter의 주요 생성자

생성자	설명
FileWriter(String name)	name이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, 이것을 매개변수로 FileWriter 객체를 생성한다.
FileWriter(String name, boolean append)	name이 의미하는 것은 파일 시스템의 실제 경로를 의미하고, append가 true이면 이어쓰기를 하고, false이면 덮어쓰기를 한다. 이 두 개의 매개변수로 FileWriter 객체를 생성한다.
FileWriter(File file)	File 객체를 이용하여 FileWriter 객체를 생성한다.
FileWriter(File file, boolean append)	이 두 개의 매개변수로 FileWriter 객체를 생성한다. append의 의미는 두 번째 생성자와 동일하다.

## 입출력 프로그래밍

### ◆ BufferedInputStream 와 BufferedOutputStream

버퍼링은 입출력 수행을 향상 시킨 기술이다.

버퍼링이란 논리적 데이터 덩어리들이 하나의 큰 물리적 입력 연산으로서 파일로부터 읽혀 버퍼로 입력하는 것을 말한다.

버퍼링을 이용하면 데이터를 읽어서 버퍼를 꽉 채운 이후 입출력을 하므로 훨씬 효율적이다.