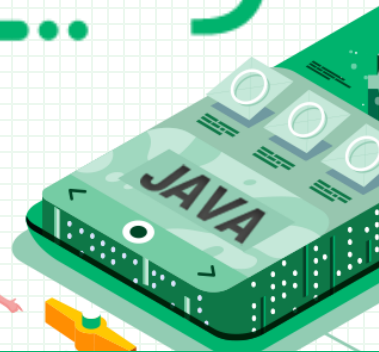




안드로이드 프로그래밍을 위한 자바기초 _...



⑪ 예외처리 및 입출력 이해하기



학습목표

- 예외처리를 이해하고 필요 시 프로그래밍에 적용시킬 수 있다.
- 입출력 스트림을 이해하고 프로그래밍에 활용할 수 있다.



학습내용

- 예외처리 이해하기
- 입출력 이해하기

⑪ 예외처리 및 입출력 이해하기

예외처리 이해하기

▶ 예외처리의 개요

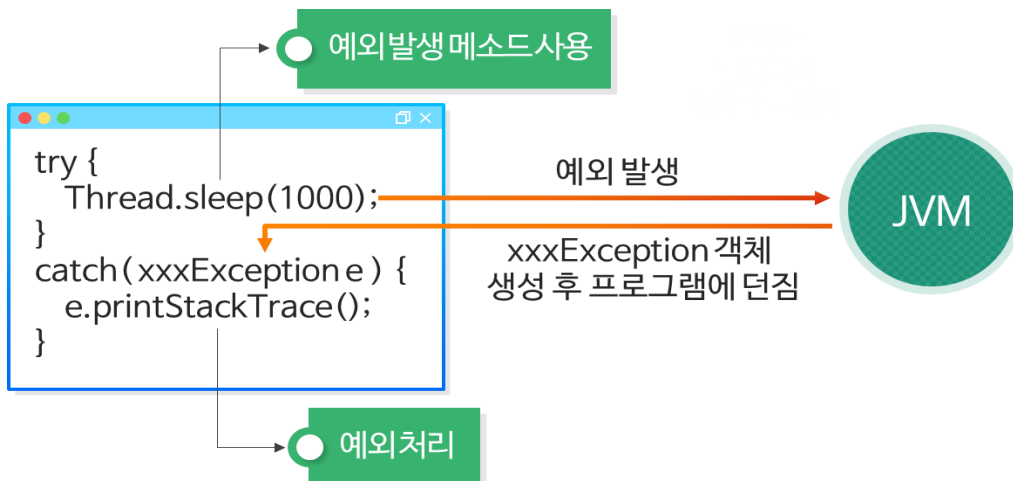
1) 예외(exception)란?

- 정상적인 프로그램 실행과정에서 발생할 수 있는 에러
- 예외를 프로그램에서 처리하면 프로그램의 안정성이 높아지고 견고해짐
- 대표적인 예외 발생 사례
 - ✓ 파일을 읽어야하는데 파일이 없는 경우
 - ✓ 네트워크 응답이 없는 경우
- 예외를 처리하기 위한 클래스
 - ✓ Throwable 클래스를 상속받은 Exception 클래스가 최상위 클래스
 - ✓ 예외 클래스 이름 : xxxException



2) 예외 발생 및 처리 순서

- (1) 프로그램 실행 중 예외가 발생
- (2) JVM이 해당 예외에 대한 xxxException 클래스의 객체를 생성하여 프로그램에 던짐
- (3) 프로그램에서 xxxException 객체를 받아서 처리



⑪ 예외처리 및 입출력 이해하기

예외처리 이해하기

▶ 예외처리 문법

1) try~catch~finally 문

```
try {
}
catch (xxxException e) {
}
catch (xxxException e) {
}
finally {
}
```

▶ try 블록 : 예외가 발생할 가능성이 있는 코드 사용

▶ catch 블록 : 예외 처리

- try 블록 내부에서 여러 종류의 예외가 발생하는 경우 여러 개 사용 가능

▶ finally 블록

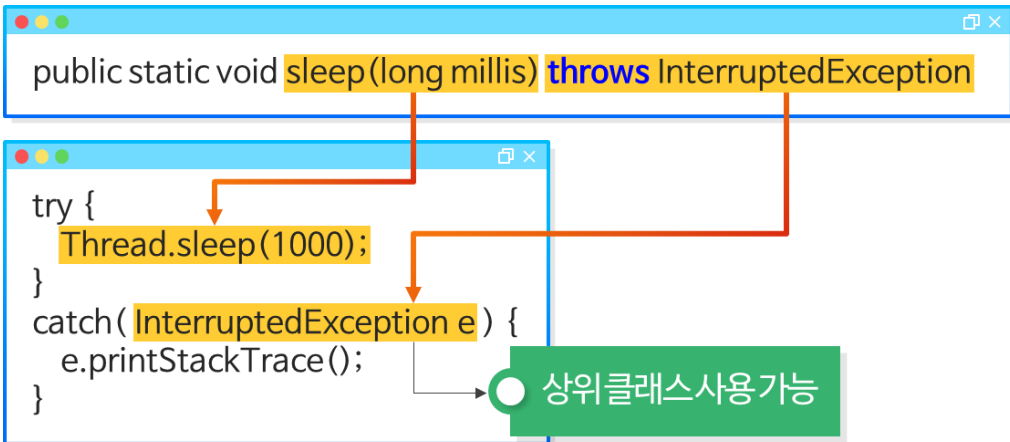
- 예외 발생 여부와 상관없이 실행, 생략해도 됨

2) throws 문

(1) 예외를 던지는 명령문

(2) throws문이 정의된 메소드

✓ 반드시 try~catch~finally 문을 사용하여야 함

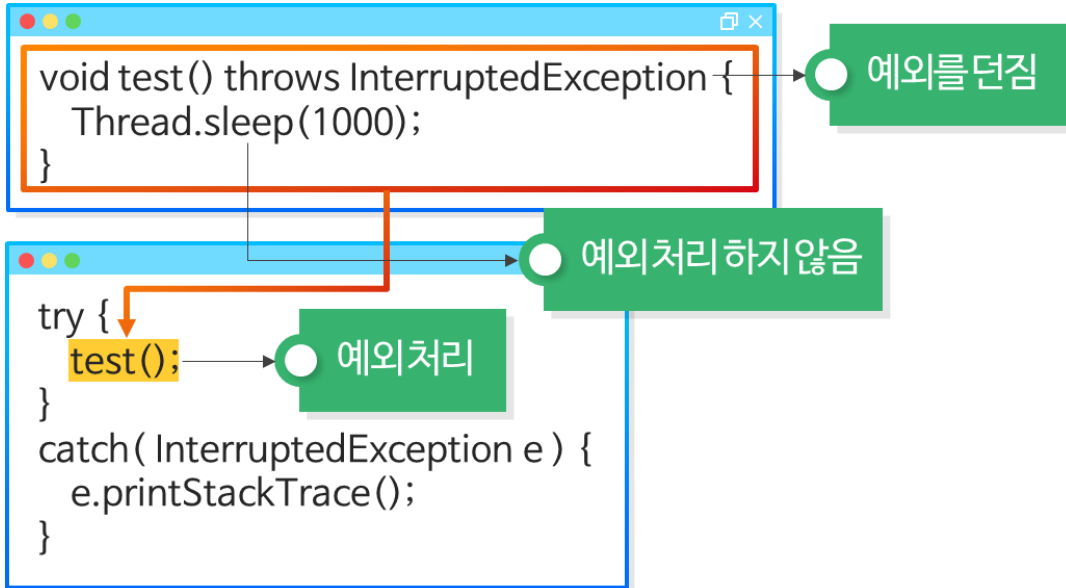


⑪ 예외처리 및 입출력 이해하기

예외처리 이해하기

(3) Throws문 사용하기

- ✓ 예외를 처리하지 않고 던지기



3) 예외 정보 출력하기

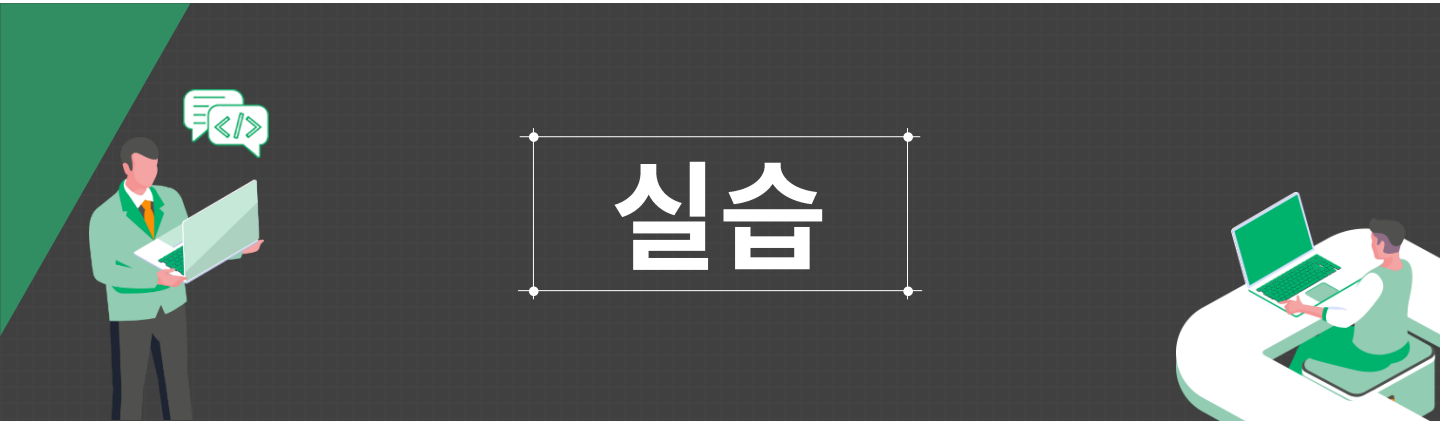
- (1) `printStackTrace()` : 예외 발생 메소드의 정보와 위치, 원인을 화면에 출력
- (2) `getMessage()` : 예외 발생 원인을 문자열로 반환

```

try {
    Thread.sleep(1000);
} catch ( InterruptedException e ) {
    e.printStackTrace();
    System.out.println(e.getMessage());
}

```

⑪ 예외처리 및 입출력 이해하기



예외처리 실습



실행 화면

```
[예외 처리를 하지 않는 경우]
Exception in thread "main"
java.lang.ArithmeticException: / by zero
    at
    exceptionclass.ExceptionClass.ex_1(ExceptionClass.java:10)
    at
    exceptionclass.ExceptionClass.main(ExceptionClass.java:37)
[예외 처리를 하는 경우]
java.lang.ArithmeticException: / by zero
    at...
```

- 소스 파일명 : [ExceptionClass.java]
- 자세한 내용은 실습 영상을 확인해보세요.

⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

➤ 입출력 스트림(stream) 이해하기

1) 입출력 스트림이란?

- 데이터의 흐름을 시냇물처럼 물 흐르듯이 처리하는 것
- 입력 : 키보드, 파일, 네트워크, 데이터베이스 등
- 출력 : 모니터, 파일, 네트워크, 데이터베이스 등

2) 입출력 스트림의 종류

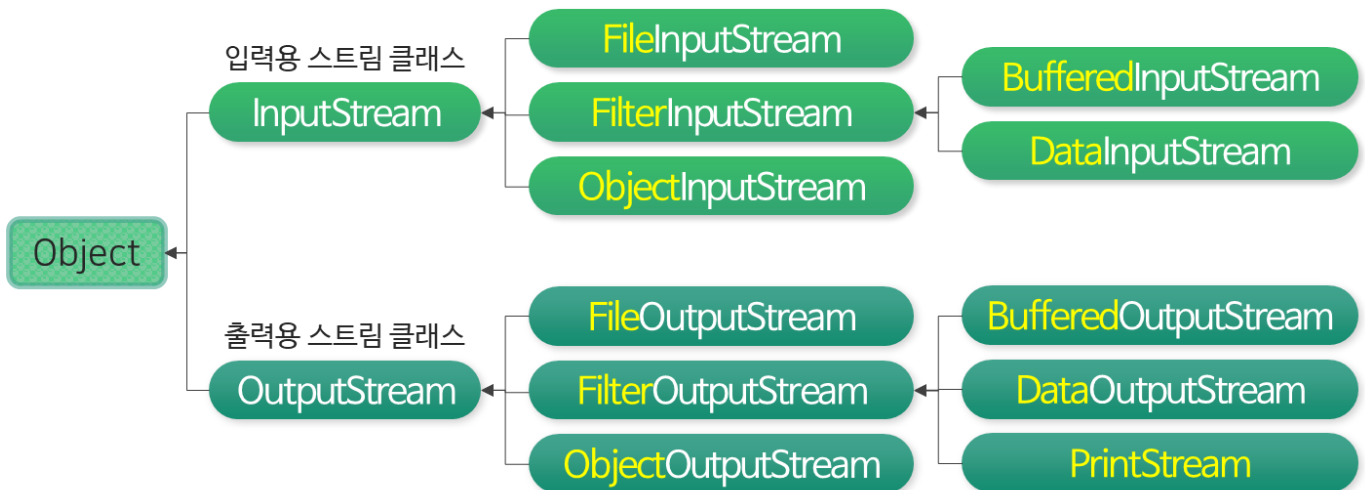
- (1) 바이트 입출력 스트림 : 바이트(byte) 단위로 입출력 하는 스트림
- (2) 문자 입출력 스트림 : 문자 단위로 입출력하는 스트림

3) 입출력 스트림 클래스

- (1) 패키지 : java.io
- (2) 입력용과 출력용 클래스로 나누어져 있음

4) 입출력 스트림 클래스 종류

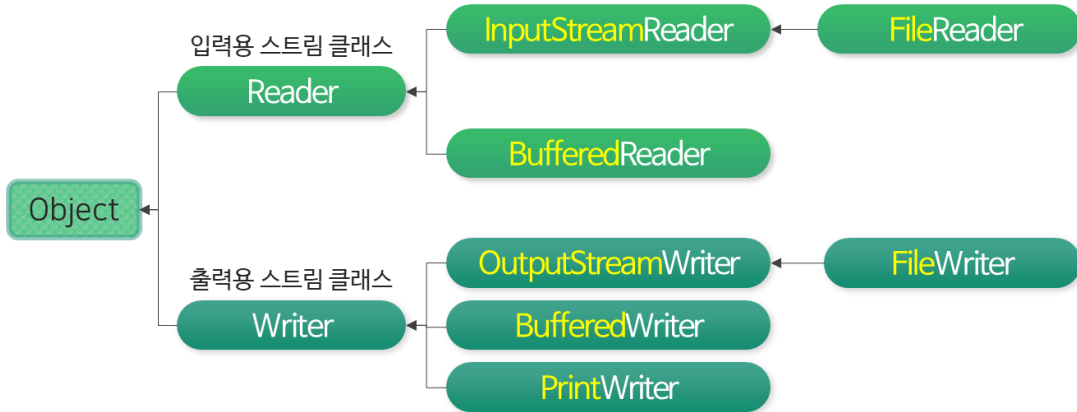
(1) 바이트 입출력 스트림 클래스



⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

(2) 문자 입출력 스트림 클래스



▶ 바이트 입출력 스트림 클래스 이해하기

1) InputStream/OutputStream 이해하기

(1) 표준 입출력 스트림

- ✓ 표준 입력 : System 클래스의 in 멤버변수(필드)
- ✓ 표준 출력 : System 클래스의 out 멤버변수(필드)

(2) 주요 메소드

✓ InputStream

int read()	<ul style="list-style-type: none"> • 1바이트를 읽어서 반환 • 반환값이 -1이면 파일 끝
int read(byte[] b)	b 배열에 읽은 데이터가 저장되며, 총 읽은 바이트수를 반환함
void close()	입력 스트림 닫기

✓ OutputStream

void write(int b)	1바이트 출력
void write(byte[] b)	b 배열의 값을 출력
void close()	출력 스트림 닫기

⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

2) FileInputStream/FileOutputStream 이해하기

(1) File 클래스

- ✓ 파일을 접근하기 위한 클래스
- ✓ 패키지 : java.io
- ✓ 주요 메소드

파일속성확인메소드

boolean canRead()	읽기 가능 여부
boolean canWrite()	쓰기 가능 여부
boolean exists()	파일 존재 여부
boolean isDirectory()	폴더인지 여부
boolean isFile()	파일인지 여부
boolean isHidden()	숨김 파일인지 여부

파일정보반환메소드

String getAbsolutePath()	절대 경로 반환
String getParent()	부모 폴더 이름 반환
String getPath()	파일 경로 반환
String getName()	파일 이름 반환
long lastModified()	마지막 수정일자 반환
long length()	파일 크기 반환
String[] list()	폴더의 파일 목록 반환

파일동작메소드

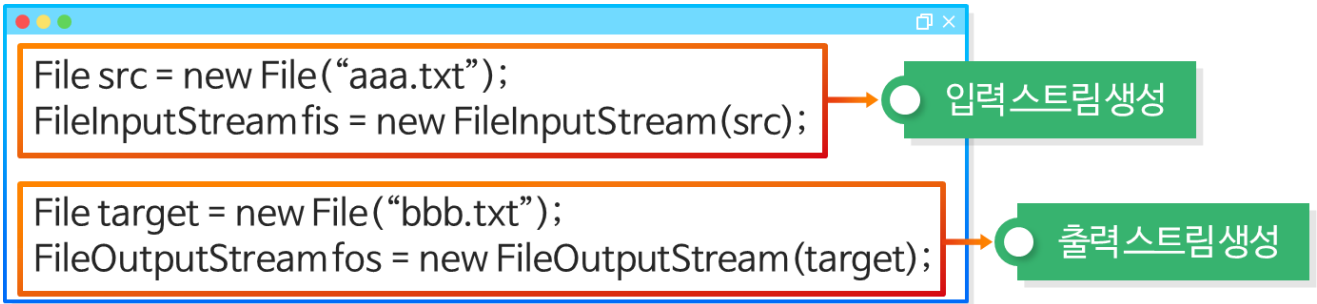
boolean delete()	파일 삭제
boolean mkdir()	폴더 생성
boolean mkdirs()	여러 개 폴더 생성
boolean renameTo(File dest)	파일 이름 변경

⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

(2) FileInputStream/FileOutputStream

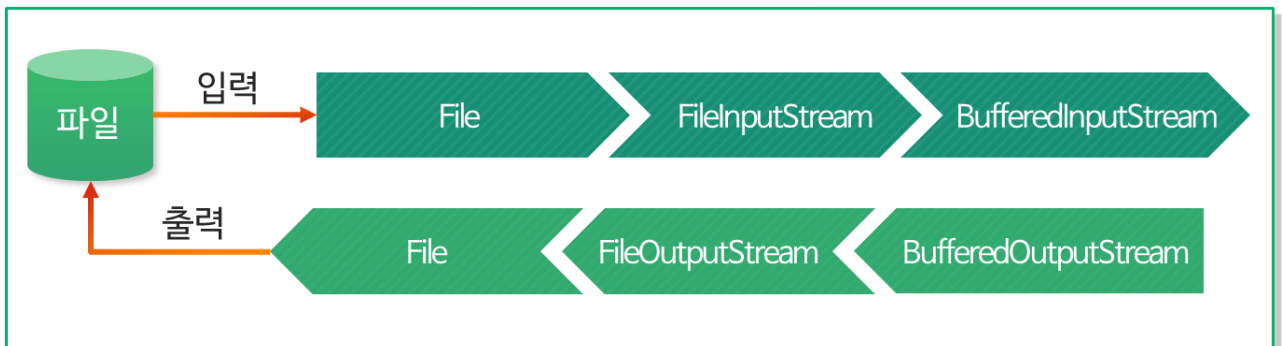
- ✓ File 클래스의 객체를 이용하여 스트림 객체를 생성함
- ✓ InputStream과 OutputStream의 하위 클래스이므로 주요 메소드는 동일함



3) BufferedInputStream/BufferedOutputStream 이해하기

- (1) 버퍼가 있는 입출력 스트림
- (2) 버퍼가 있으면 데이터를 처리하는 다양한 메소드가 존재
- (3) 데이터 출력 시 close() 메소드로 스트림을 닫기 전에 버퍼에 있는 데이터를 출력하는 flush() 메소드를 호출해야 함
- (4) 기본 입출력 스트림을 연결하여 사용

버퍼입출력스트림사용



⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

4) DataInputStream/DataOutputStream 이해하기

- (1) 기본 자료형을 입출력 할 수 있는 스트림
- (2) 기본 자료형 각각에 맞는 readXXX() 메소드와 writeXXX() 메소드가 존재함

기본 자료형	read() 메소드	write() 메소드
boolean	readBoolean()	writeBoolean()
char	readChar()	writeChar()
byte	readByte()	writeByte()
short	readShort()	writeShort()
int	readInt()	writeInt()
long	readLong()	writeLong()
float	readFloat()	writeFloat()
double	readDouble()	writeDouble()
문자열	readUTF()	writeUTF()

5) ObjectInputStream/ObjectOutputStream 이해하기

- (1) 객체 직렬화가 가능한 클래스의 객체를 입출력 할 수 있는 스트림
- (2) 객체 직렬화(object serialization) 가능 클래스
 - ✓ Serializable 또는 Externalizable 인터페이스를 구현한 클래스
- (3) 자바에서 제공하는 대부분의 클래스들은 객체 직렬화 가능

```
public final class String extends Object implements Serializable
```

(4) 주요 메소드

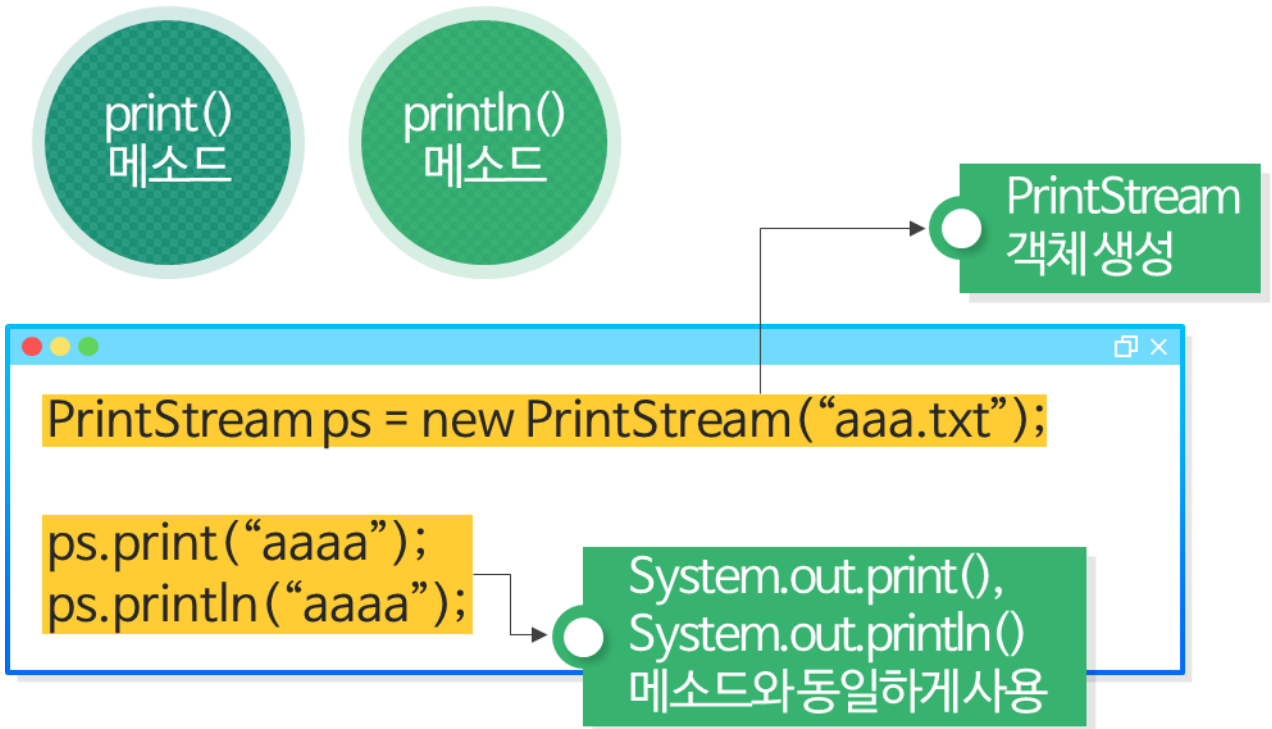
- ✓ 객체 입력 : readObject()
- ✓ 객체 출력 : writeObject()

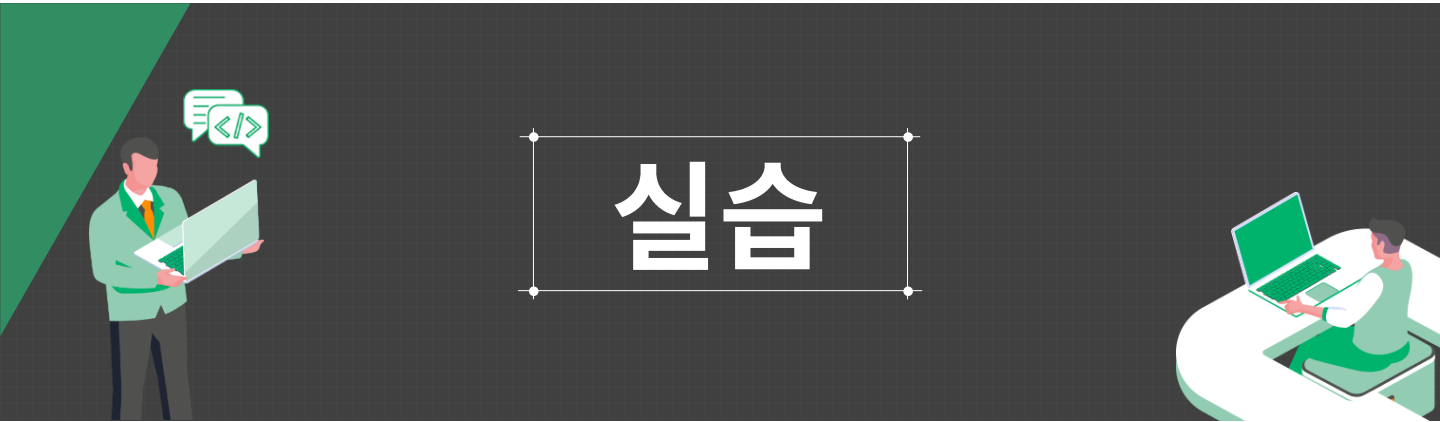
⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

6) PrintStream 이해하기

- (1) 텍스트 기반으로 출력하는 스트림 클래스
- (2) 주요 메소드





바이트 입출력 스트림 실습



실행 화면

```
[InputStrem/OutputStream 실습]
abcdef
abcdef
123456
123456
x
count = 16
[File 실습]
File Name => .project
Parh = E:\JAVA\11회차\byteio\project
Date = Sat Jul 25 10:47:38 GMT+09:00 2020
Size = 382
Read = true...
```

- 소스 파일명 : [ByteIO.java]
- 자세한 내용은 실습 영상을 확인해보세요.

⑪ 예외처리 및 입출력 이해하기

입출력 이해하기

➤ 문자 입출력 스트림 클래스 이해하기

1) 바이트 입출력 스트림 클래스와 비교

입력스트림대응클래스

설명	바이트 스트림 클래스	문자 스트림 클래스
기본 입력 스트림	InputStream	InputStreamReader
파일 입력 스트림	FileInputStream	FileReader
필터 입력 스트림	FilterInputStream	FilterReader
버퍼가 있는 입력 스트림	BufferedInputStream	BufferedReader

출력스트림대응클래스

설명	바이트 스트림 클래스	문자 스트림 클래스
기본 출력 스트림	OutputStream	OutputStreamWriter
파일 출력 스트림	FileOutputStream	FileWriter
버퍼가 있는 출력 스트림	BufferedOutputStream	BufferedWriter
표준 출력 스트림	PrintStream	PrintWriter

대응클래스없는경우

설명	바이트 스트림 클래스	문자 스트림 클래스
기본 자료형 입력 스트림	DataInputStream	-
객체 입력 스트림	ObjectInputStream	-
기본 자료형 출력 스트림	DataOutputStream	-
객체 출력 스트림	ObjectOutputStream	-

⑪ 예외처리 및 입출력 이해하기

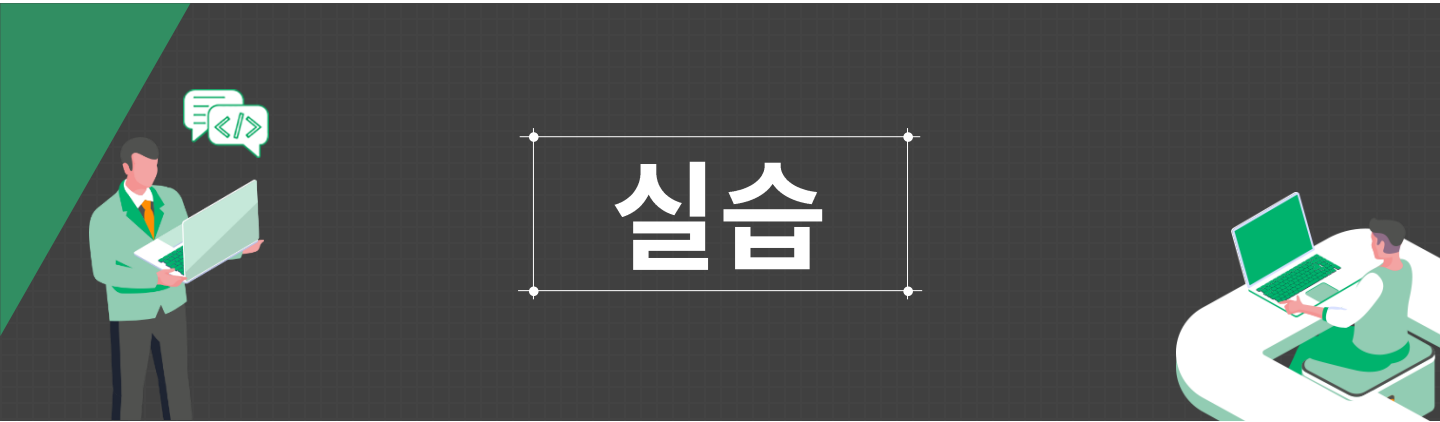
입출력 이해하기

2) 주요 메소드

- (1) 바이트 입출력 스트림에 있는 메소드는 동일하게 사용
- (2) 유용한 메소드

클래스	메소드	설명
BufferedReader	String readLine()	텍스트 한 줄 읽어오기
BufferedWriter	void newLine()	새로운 줄 바꿈 출력하기

⑪ 예외처리 및 입출력 이해하기



실습

문자 입출력 스트림 실습



실행 화면

```
[InputStreamReader/OutputStreamWriter 실습]
abcdef
123456
x
count = 16
abcdef
123456
[FileReader/FileWriter 실습 : 파일 복사 프로그램]
Copy Size : 382byte
[BufferedReader/BufferedWriter 실습 : 파일 복사
프로그램]
Line=17
```

- 소스 파일명 : [CharIO.java]
- 자세한 내용은 실습 영상을 확인해보세요.



정리하기

■ 예외처리 이해하기

- 예외(exception)
 - 정상적인 프로그램 실행과정에서 발생할 수 있는 에러
 - 예외를 프로그램에서 처리하면 프로그램의 안정성이 높아지고 견고해 짐
 - 대표적인 예외 발생 사례
 - 파일을 읽어야 하는데 파일이 없는 경우, 네트워크 응답이 없는 경우
 - 예외를 처리하기 위한 클래스
 - Throwable 클래스를 상속받은 Exception 클래스가 최상위 클래스
 - 예외 클래스 이름 : xxxException
- 예외 발생 및 처리 순서
 - 프로그램 실행 중 예외가 발생
 - JVM이 해당 예외에 대한 xxxException 클래스의 객체를 생성하여 프로그램에 던짐
 - 프로그램에서 xxxException 객체를 받아서 처리



정리하기

■ 예외처리 이해하기

- try~catch~finally 문
 - try 블록 : 예외가 발생할 가능성이 있는 코드
 - catch 블록 : 예외 처리
 - finally 블록 : 예외 발생 여부와 상관없이 실행, 생략해도 됨
- throws 문
 - 예외를 던지는 명령문
 - throws 문이 정의된 메소드
- 예외 정보 출력하기
 - printStackTrace() : 예외 발생 메소드의 정보와 위치, 원인을 화면에 출력
 - getMessage() : 예외 발생 원인을 문자열로 반환



정리하기

■ 입출력 이해하기

- 입출력 스트림(stream)
 - 데이터의 흐름을 시냇물처럼 물 흐르듯이 처리하는 것
 - 입출력 스트림의 종류
 - 바이트 입출력 스트림 : 바이트(byte) 단위로 입출력하는 스트림
 - 문자 입출력 스트림 : 문자 단위로 입출력하는 스트림
 - 입출력 스트림 클래스
 - 패키지 : java.io
 - 입력용과 출력용 클래스로 나누어져 있음
- InputStream/OutputStream
 - 표준 입출력 스트림
 - 표준 입력 : System 클래스의 in 멤버변수(필드)
 - 표준 출력 : System 클래스의 out 멤버변수(필드)



정리하기

■ 입출력 이해하기

- FileInputStream/FileOutputStream
 - File 클래스 : 파일을 접근하기 위한 클래스
 - FileInputStream/FileOutputStream : File 클래스의 객체를 이용하여 스트림 객체를 생성함
- BufferedInputStream/BufferedOutputStream
 - 버퍼가 있는 입출력 스트림
- DataInputStream/DataOutputStream
 - 기본 자료형을 입출력 할 수 있는 스트림
- ObjectInputStream/ObjectOutputStream
 - 객체 직렬화가 가능한 클래스의 객체를 입출력 할 수 있는 스트림
- PrintStream
 - 텍스트 기반으로 출력하는 스트림 클래스