

K G 아 이 티 뱅 크

J A V A

J A V A

Bridged Stream Class

Bridged Stream Class

❖ Input/Output Stream을 다른 스트림으로 변환하는 클래스

- File Stream의 입출력은 기본적으로 Byte단위 입출력을 사용
- Data Stream은 이를 지정한 형식으로 처리하는 기능을 추가
- Bridged Stream Class는 이러한 기반을 완전히 변형함

❖ Reader/Writer 추상 클래스를 상속받아 문자 입출력을 함

- 이 과정에서 필요할 경우 인코딩을 지정하여 사용도 가능함

```
import java.io.Writer; // 추상 클래스이며 부모클래스
import java.io.Reader;
import java.io.OutputStreamWriter; // 구현 클래스
import java.io.InputStreamReader;
....
// 간결함 / 가독성을 위해 업캐스팅으로 사용함
Writer charWriter = new OutputStreamWriter(ostream);
// 생성시 예외처리는 Stream 객체 생성시에 수행함
Reader charReader = new InputStreamReader(istream);
```

Bridged Stream Class

❖ Reader/Writer Class는 오로지 문자 입출력만을 수행함

- ① File Stream의 read/write 메서드는 1 Byte 입출력을 함
- ② Reader/Writer의 read/write는 2 Byte 문자 입출력을 함
- ③ 자료형은 int이나 무부호 2 Byte를 위한 것이니 주의
- ④ 처리하는 값의 일관성을 위해 인코딩은 지정하는게 좋음

```
try {  
    // 인코딩방식을 지정하지 않으면 자동선정됨  
    Writer fw = new OutputStreamWriter(ostream, "UTF-8");  
    fw.write("문자열을 전달하여 처리 가능!");  
    fw.close(); // close는 항상 처리해줘야 함  
    Reader fr = new InputStreamReader(istream, "UTF-8");  
    System.out.println((char)fr.read());  
    // 형변환을 적용하며, 더이상 읽을 수 없으면 -1을 반환함  
    fr.close();  
}  
catch(IOException e) { System.out.println("ERROR"); }
```

Buffered Stream Class

Buffered Stream Class

❖ I/O 처리속도를 높여주는 Buffer를 추가하는 클래스

- Buffer : 충격/불편감을 줄여주는 역할을 수행하는 것
- 컴퓨터에서는 이를 임시저장공간이라고 함

❖ 지나치게 빈번한 I/O처리를 줄여 속도를 올리는 것

- 저장공간은 속도가 서로 많이 달라 "충격"이 발생함
- "충격"이 발생하는 빈도를 줄여 속도를 올리는 것

```
import java.io.BufferedInputStream; // Stream에 가능
import java.io.BufferedOutputStream;
import java.io.BufferedReader; // Reader에만 가능
import java.io.BufferedWriter; // Writer에만 가능
// 생성시에는 문제가 발생안함
BufferedInputStream buffered_is =
    new BufferedInputStream(istream);
// 이 클래스들은 전부 기능추가에 속하는 wrapper클래스
BufferedReader dis =
    new BufferedReader(readerObject);
```

Buffered Stream Class

❖ BufferedInput/Output은 close 외에 flush가 존재함

- ① flush : 버퍼의 내용을 파일에 기록하고 버퍼를 비움
- ② buffer가 가득차면 자동으로 호출되어 기록을 수행함
- ③ flush 메서드는 특수상황 외에는 잘 호출하지 않음
- ④ 이 클래스는 기능추가를 담당하는 wrapper이니 주의

// 예외처리는 당연히 필수로 요구하니 주의

```
FileOutputStream fos = new FileOutputStream("FILE");  
OutputStream bOut = new BufferedOutputStream(fos);  
bOut.write(65); // buffer에 보관됨  
bOut.flush();  // 파일에 기록하고 buffer를 비움  
bOut.close();  // close시 flush가 자동으로 호출됨
```

```
FileInputStream fis = new FileInputStream("FILE");  
InputStream bIn = new BufferedInputStream(fis);  
byte data = bIn.read(); // 자동으로 Buffer를 채움  
bIn.close(); // buffer에 기록된 내용이 모두 제거됨
```

Buffered Stream Class

❖ BufferedReader/Writer는 Stream의 Buffer와 다름

- ① Buffered Stream은 바이트 단위 I/O에 버퍼를 추가함
- ② Buffered Reader/Writer는 문자단위 I/O에 추가해줌
- ③ 작업 빈도에 따라 다르지만 이중 버퍼는 피해야 함
- ④ 둘의 성능차이는 환경에 따라 다르니 측정을 해야 함

```
// Byte 단위 버퍼는 InputStream에 Buffer를 적용함
InputStream is = new FileInputStream("SAMPLE.txt");
is = new BufferedInputStream(is);
// 단순 Reader이지만 이미 Buffered가 적용되어 있음
Reader freader = new InputStreamReader(is, "UTF-8");
```

```
// 문자 단위 버퍼는 Reader에 buffer를 적용
InputStream is = new FileInputStream("SAMPLE.txt");
// is는 버퍼가 없고, freader에만 버퍼가 존재함
Reader freader = new InputStreamReader(is, "UTF-8");
freader = new BufferedReader(freader);
```