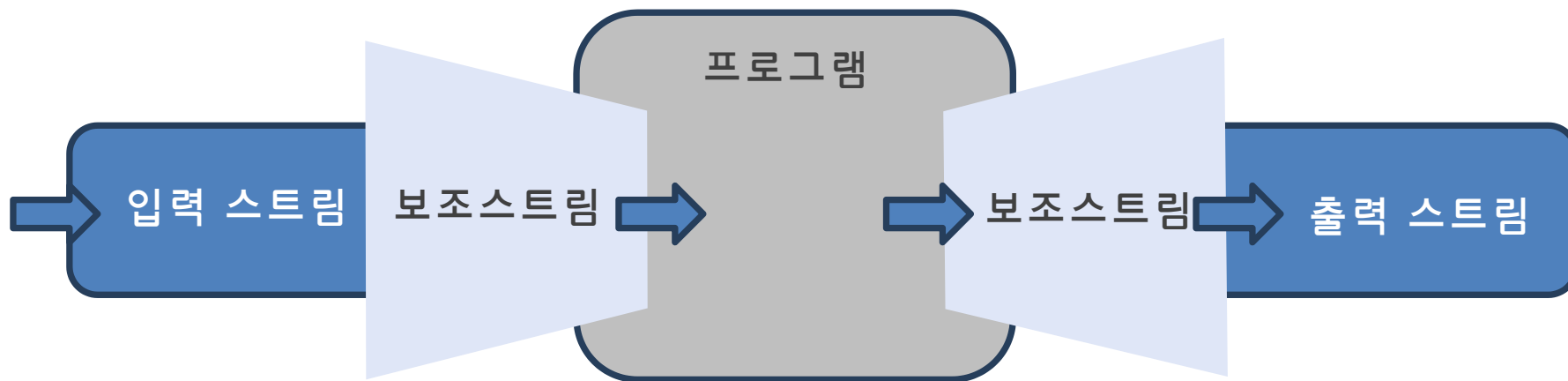


01. 보조 입출력 스트림

■ 보조 스트림

- 다른 스트림과 연결이 되어 여러 편리한 기능을 제공
- 단독으로는 사용 불가, 기본 입출력 소스와 함께 사용

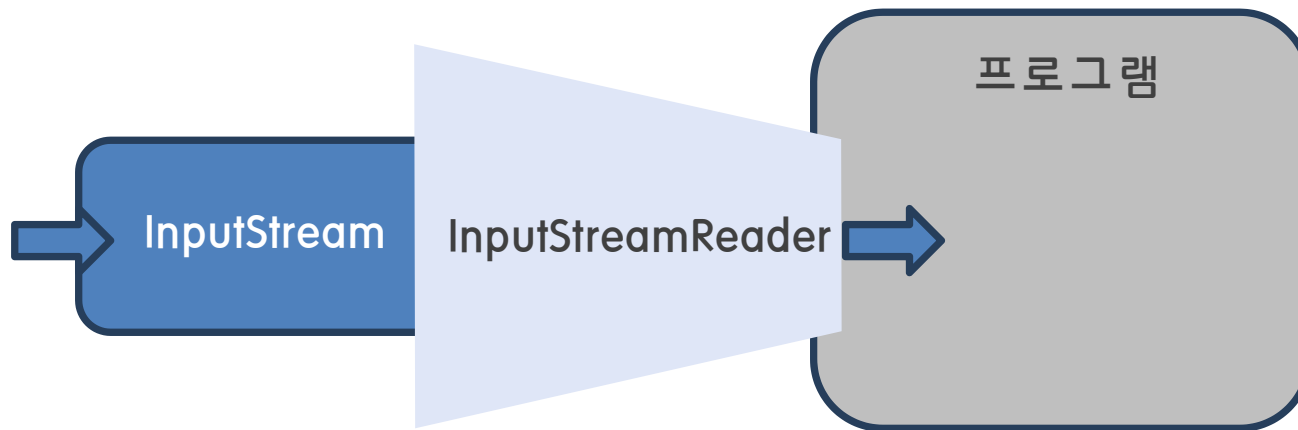


02. 보조 스트림 연결하기

■ 연결 방법 기본

```
보조스트림 변수 = new 보조스트림(연결스트림)
```

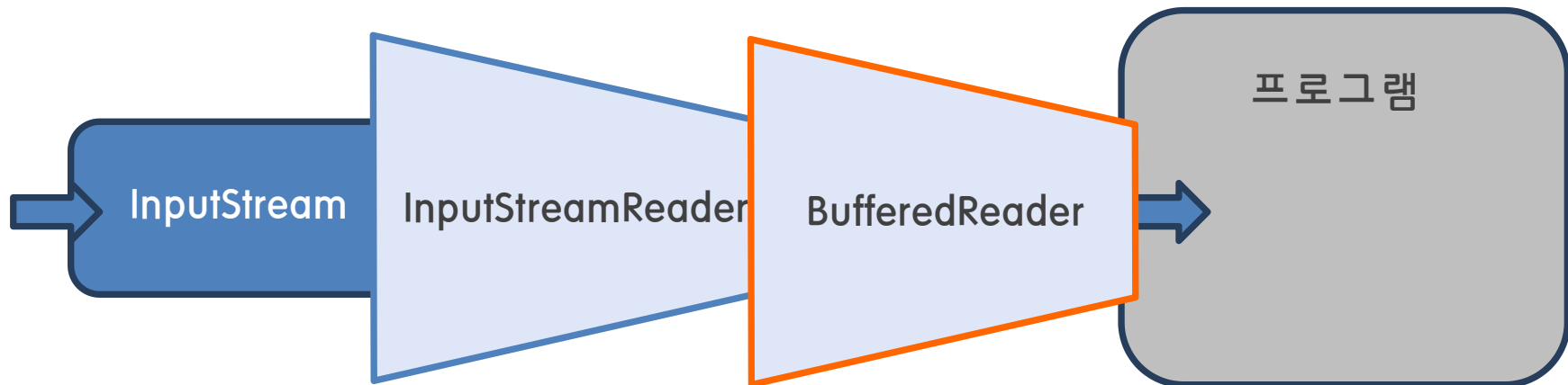
```
InputStream is = ...;  
InputStreamReader reader = new InputStreamReader(is);
```



02. 보조 스트림 연결하기

■ 성능향상 보조스트림 연결

```
InputStream is = ...;  
InputStreamReader reader = new InputStreamReader(is);  
BufferedReader br = new BufferedReader(reader);
```



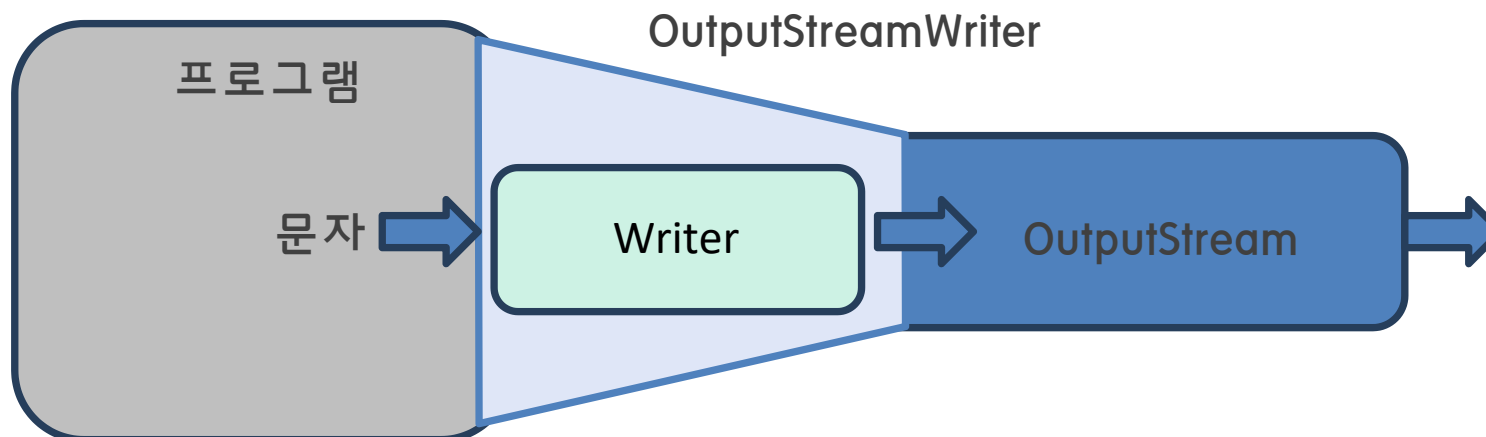
03. 문자 변환 보조 스트림 연결하기

■ OutputStreamWriter

- 바이트 기반 출력스트림에 연결되어 문자출력 스트림인 Writer로 변환

```
Writer writer = new OutputStreamWriter(바이트 기반 출력 스트림) ;
```

```
FileOutputStream fos = new FileOutputStream(" C:/Temp/test.txt" )  
Writer writer = new OutputStreamWriter(fos);
```



03. 문자 변환 보조 스트림 연결하기

■ InputStreamReader

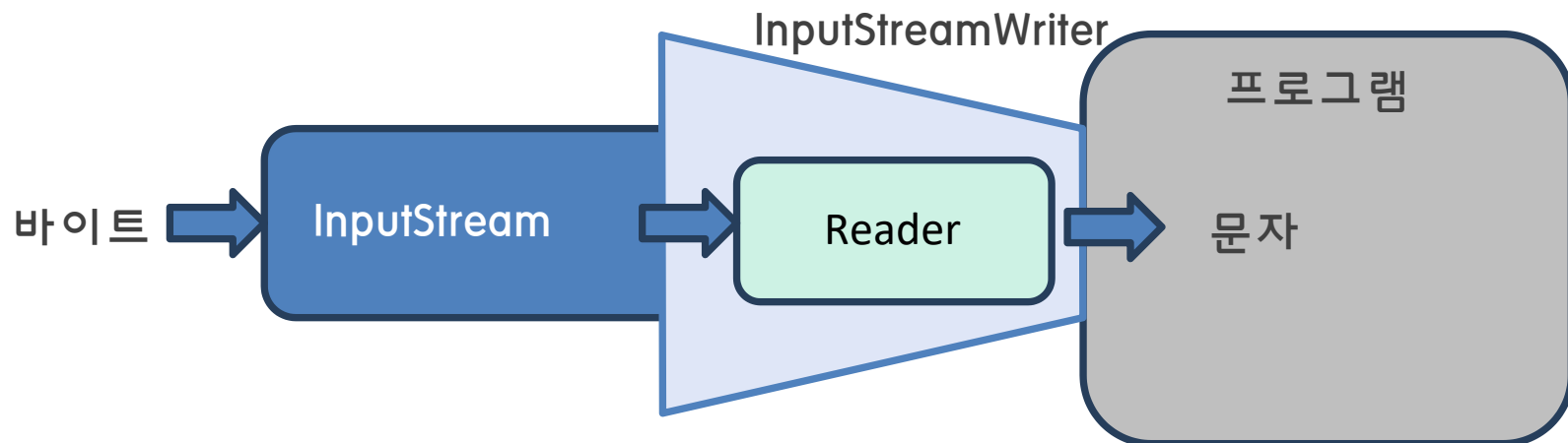
[예제]

[CC01CharacterConvertStreamExample.java](#)

- 바이트 기반 입력 스트림에 연결되어 문자 입력 스트림인 Reader 로 변환

```
Reader reader = new InputStreamReader(바이트 기반 입력 스트림);
```

```
FileInputStream fis = new FileInputStream(" C:/Temp/test1.txt" );  
Reader reader = new InputStreamReader(fis);
```

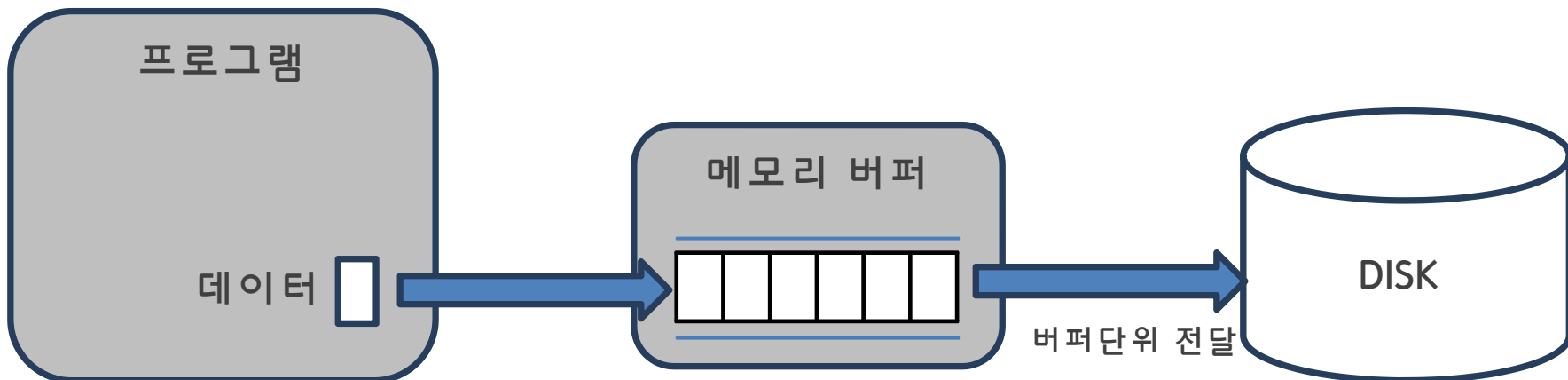


04. 성능 향상 보조 스트림

■ 버퍼 공간

■ 버퍼 공간이란

- 입출력 중간의 버퍼라는 메모리 공간이 존재하고 해당공간 단위로 데이터 입출력이 진행
- 효율적인 데이터 전달위한 공간



04. 성능 향상 보조 스트림

■ BufferedOutputStream 과 BufferedWriter

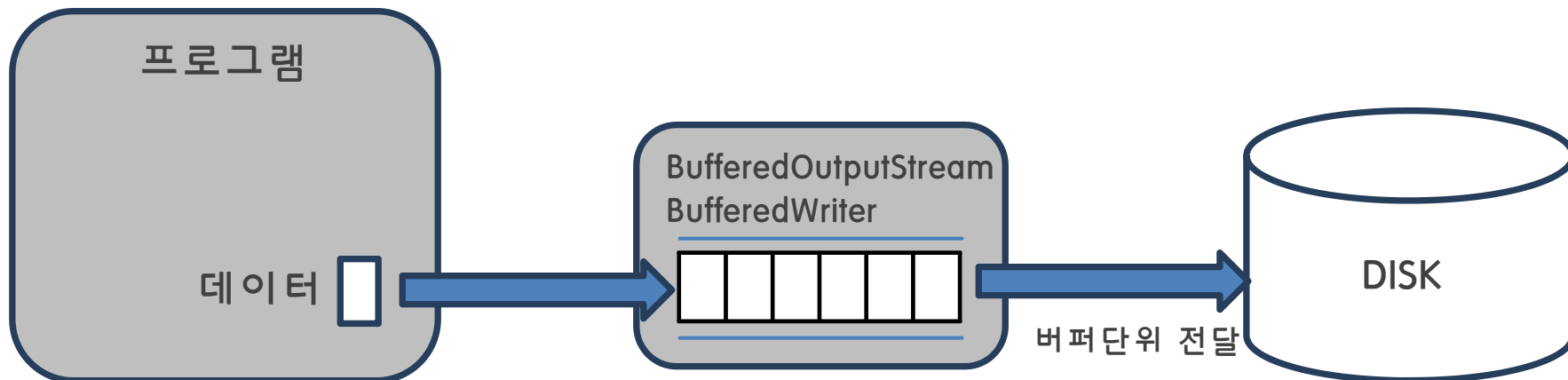
■ BufferedOutputStream

- 바이트 기반 출력 스트림에 연결되어 버퍼를 제공하는 보조 스트림

■ BufferedWriter

- 문자 기반 출력 스트림에 연결되어 버퍼를 제공하는 보조 스트림

```
BufferedOutputStream bos = new BufferedOutputStream(바이트 기반 출력 스트림);  
BufferedWriter bw = new BufferedWriter(문자 기반 출력 스트림);
```



04. 성능 향상 보조 스트림

■ BufferedInputStream 과 BufferedReader

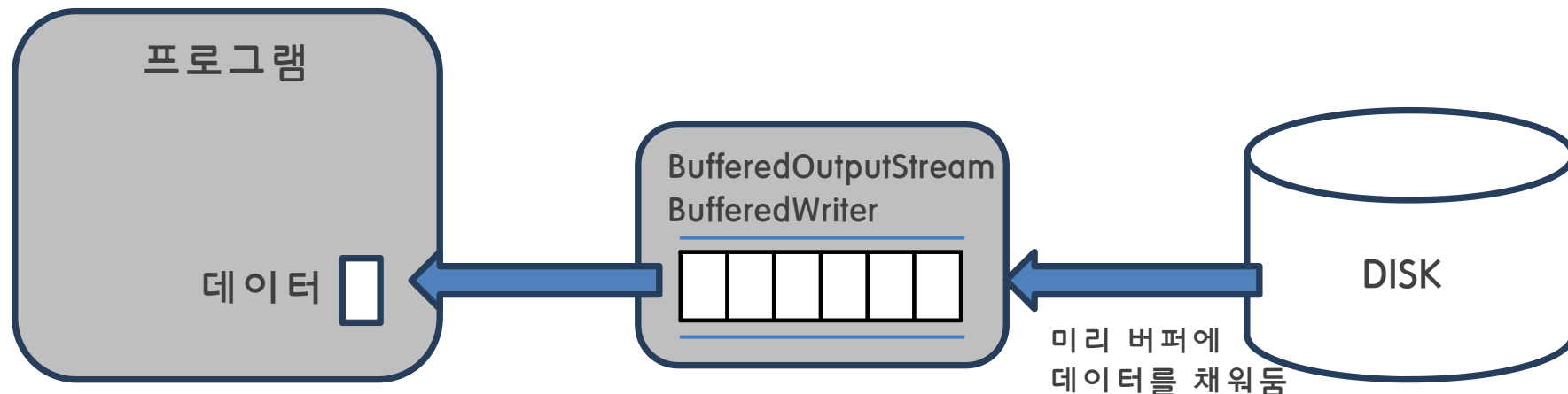
■ BufferedInputStream

- 바이트 기반 입력 스트림에 연결되어 버퍼를 제공하는 보조 스트림

■ BufferedReader

- 문자 기반 입력 스트림에 연결되어 버퍼를 제공하는 보조 스트림

```
BufferedInputStream bis = new BufferedInputStream(바이트 기반 입력 스트림);  
BufferedReader br = new BufferedReader(문자 기반 입력 스트림);
```



05. 기본 타입 입출력 스트림

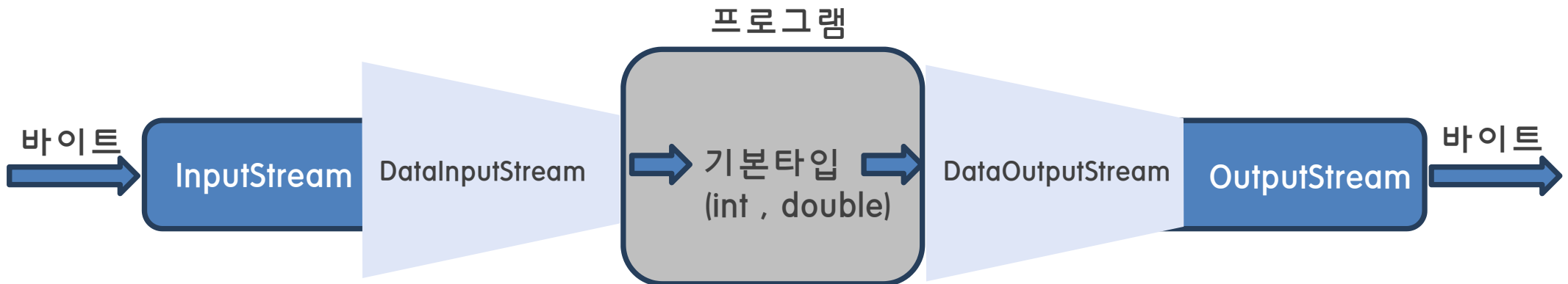
[예제]

C04DataInputStreamExample

■ DataInputStream / DataOutputStream

- 기본 타입인 boolean, char, short, int, long, float, double을 입출력 할 수 있다

```
DataInputStream dis = new DataInputStream(바이트 기반 입력 스트림);  
DataOutputStream dos = new DataOutputStream(바이트 기반 출력 스트림);
```



06. 프린트 보조 스트림

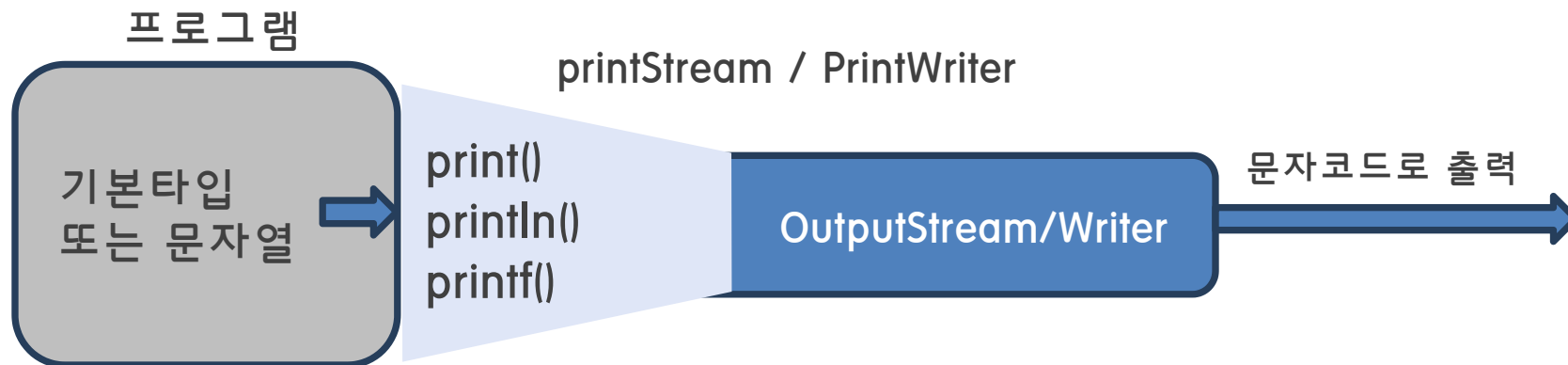
[예제]

C05PrintStreamExample.java

■ PrintStream / PrintWriter

- print() , println()가 지고 있는 보조 스트림

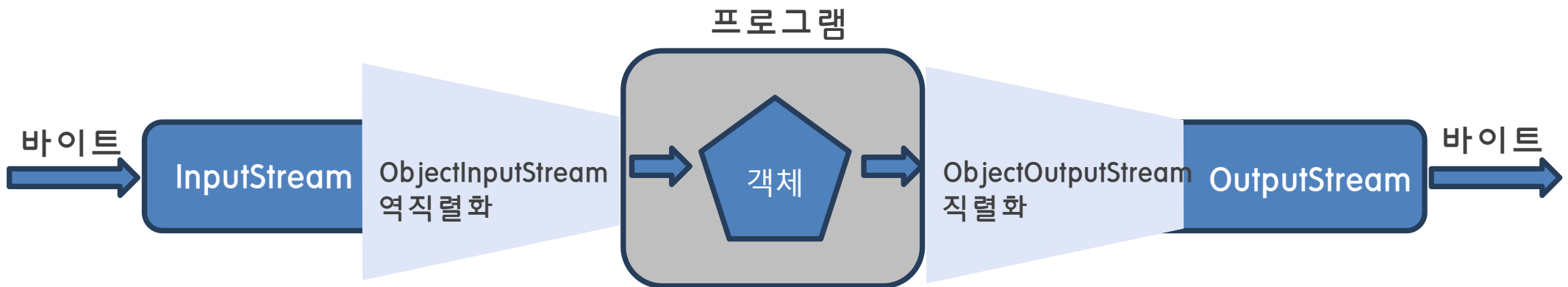
```
PrintStream ps = new PrintStream(바이트 기반 출력 스트림);  
PrintWriter pw = new PrintWriter(문자 기반 출력 스트림);
```



07. 객체 입출력 보조 스트림

■ ObjectOutputStream / ObjectInputStream

- 객체를 직렬화/역직렬화 하여 전달



07. 객체 입출력 보조 스트림

■ ObjectOutputStream / ObjectInputStream

■ 기본 사용법

```
ObjectInputStream ois = new ObjectInputStream(바이트 기반 스트림);  
ObjectOutputStream oos = new ObjectOutputStream(바이트 기반 출력 스트림);
```

■ 직렬화

```
oos.writeObject(객체);
```

■ 역직렬화

```
객체타입 변수 = (객체타입)ois.readObject();
```

■ 직렬화 구현 인터페이스 사용

```
public class XXX implements Serializable { ...}
```