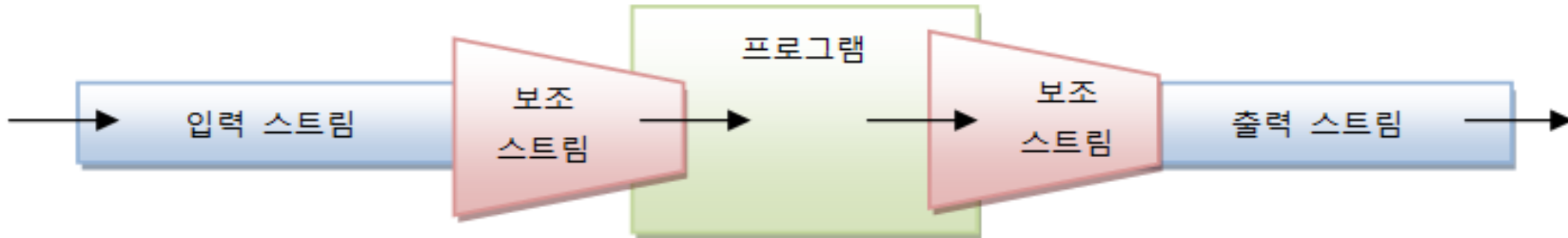


보조 스트림

보조 스트림

❖ 보조 스트림

- 다른 스트림과 연결 되어 여러 가지 편리한 기능을 제공하는 스트림
 - 문자 변환, 입출력 성능 향상, 기본 데이터 타입 입출력, 객체 입출력 등의 기능을 제공



보조 스트림

❖ 보조 스트림

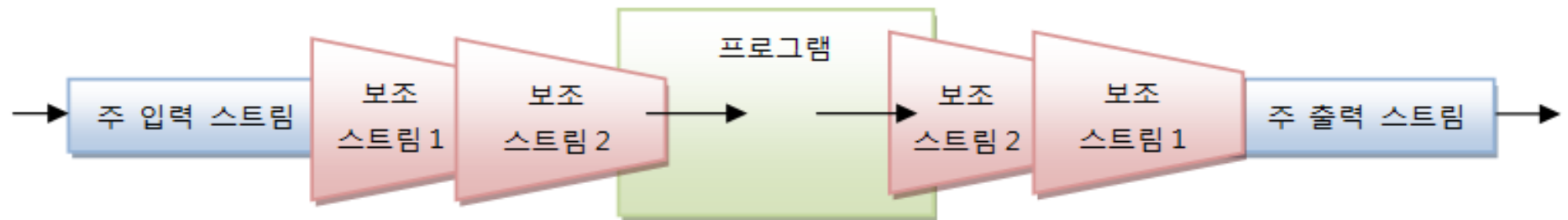
- 보조 스트림 생성

```
보조스트림 변수 = new 보조스트림(연결스트림)
```

- 보조 스트림 체인 - 다른 보조 스트림과 연결되어 역할 수행

```
InputStream is = System.in;
```

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



```
InputStream is = System.in;
```

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

```
BufferedReader br = new BufferedReader(reader);
```

보조 스트림

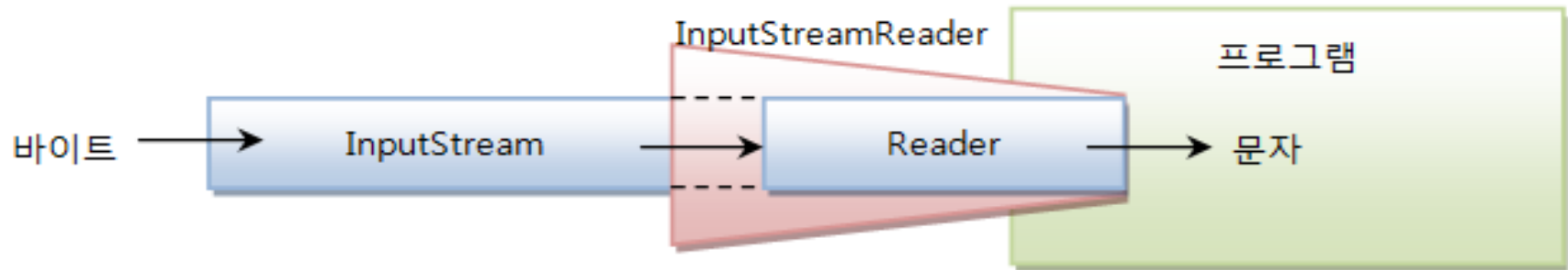
❖ 문자 변환 보조 스트림

- 소스 스트림이 바이트 기반 스트림이지만 데이터가 문자일 경우 사용
 - Reader와 Writer는 문자 단위로 입출력 - 바이트 기반 스트림보다 편리
 - 문자셋의 종류를 지정할 수 있기 때문에 다양한 문자 입출력 가능

보조 스트림

❖ 문자 변환 보조 스트림

○ InputStreamReader



```
Reader reader = new InputStreamReader(바이트입력스트림);
```

```
InputStream is = System.in;
```

```
Reader reader = new InputStreamReader(is);
```

```
FileInputStream fis = new FileInputStream("C:/Temp/file.txt");
```

```
Reader reader = new InputStreamReader(fis);
```

❖ 콘솔에서 한글 입력받기: InputStreamReaderExample.java

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;

public class InputStreamReaderExample {
    public static void main(String[] args) throws Exception {
        InputStream is = System.in;
        Reader reader = new InputStreamReader(is);

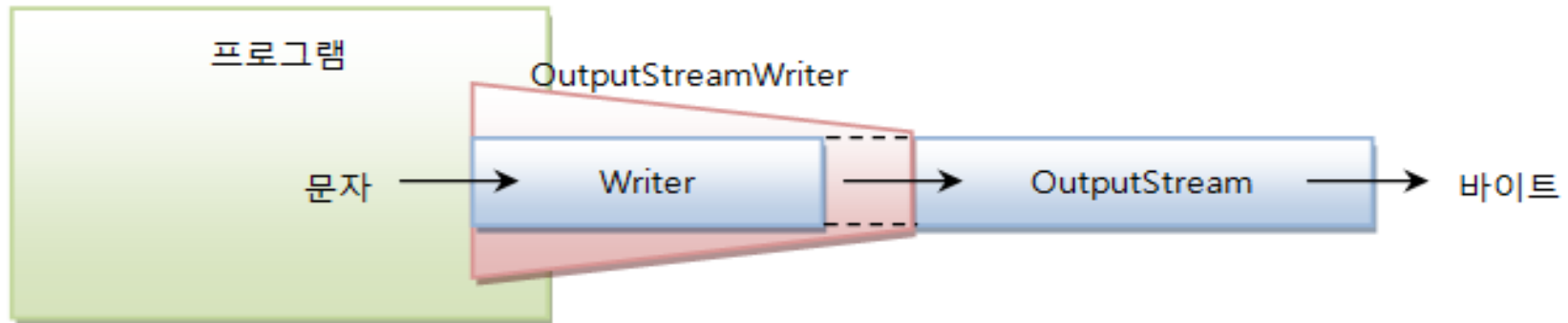
        int readCharNo;
        char[] cbuf = new char[100];
        while ((readCharNo=reader.read(cbuf)) != -1) {
            String data = new String(cbuf, 0, readCharNo);
            System.out.println(data);
        }

        reader.close();
    }
}
```

보조 스트림

❖ 문자 변환 보조 스트림

○ OutputStreamWriter



```
Writer writer = new OutputStreamWriter(바이트출력스트림);
```

```
FileOutputStream fos = new FileOutputStream("C:/Temp/file.txt");
```

```
Writer writer = new OutputStreamWriter(fos);
```

❖ 파일로 출력하기: OutputStreamWriterExample.java

```
import java.io.FileOutputStream;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.io.Reader;
import java.io.Writer;

public class OutputStreamWriterExample {
    public static void main(String[] args) throws Exception {
        FileOutputStream fos = new FileOutputStream("C:/Temp/file.txt");
        Writer writer = new OutputStreamWriter(fos);

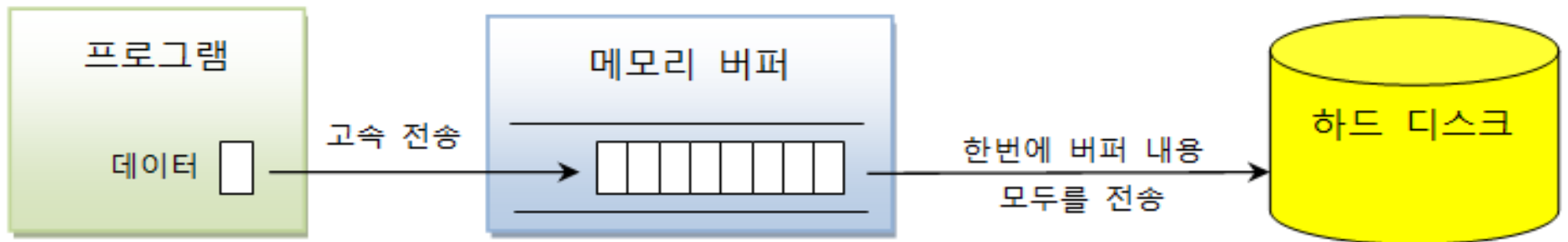
        String data = "바이트 출력 스트림을 문자 출력 스트림으로 변환";
        writer.write(data);

        writer.flush();
        writer.close();
        System.out.println("파일 저장이 끝났습니다.");
    }
}
```


보조 스트림

❖ 성능 향상 보조 스트림

- 입출력 성능에 영향을 미치는 입출력 소스
 - 하드 디스크
 - 느린 네트워크
- 버퍼를 이용한 해결
 - 입출력 소스와 직접 작업하지 않고 버퍼(buffer)와 작업 - 실행 성능 향상

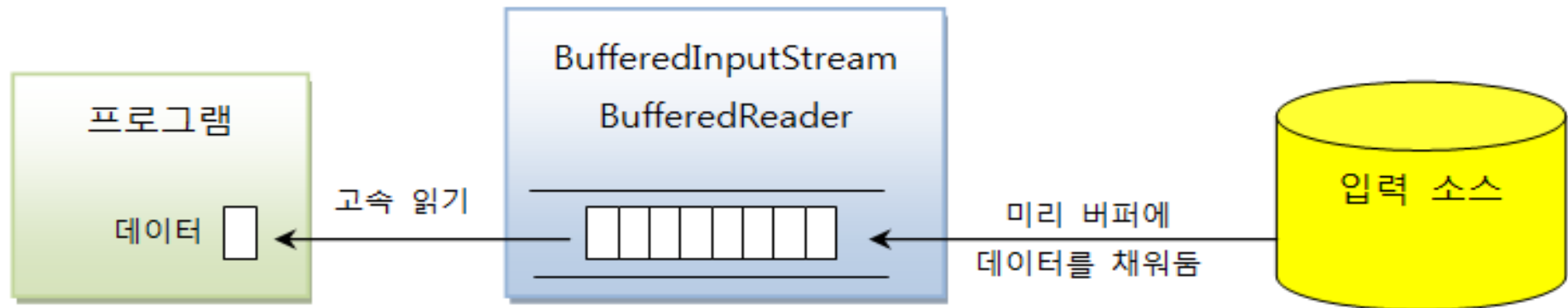


- 프로그램은 쓰기 속도 향상
- 버퍼 차게 되면 데이터를 한꺼번에 하드 디스크로 보내 출력 횟수를 줄여줌

보조 스트림

❖ 성능 향상 보조 스트림

- BufferedInputStream, BufferedReader



```
BufferedInputStream bis = new BufferedInputStream(바이트입력스트림);
```

```
BufferedReader br = new BufferedReader(문자입력스트림);
```

❖ 읽기 성능 테스트: BufferedInputStreamExample.java

```
import java.io.BufferedInputStream;
import java.io.FileInputStream;

public class BufferedInputStreamExample {
    public static void main(String[] args) throws Exception {
        long start = 0;
        long end = 0;

        FileInputStream fis1 = new FileInputStream("C:/temp/forest.jpg");
        start = System.currentTimeMillis();
        while(fis1.read() != -1) {}
        end = System.currentTimeMillis();
        System.out.println("사용하지 않았을 때: " + (end-start) + "ms");
        fis1.close();
    }
}
```

❖ 읽기 성능 테스트: BufferedInputStreamExample.java

```
FileInputStream fis2 = new FileInputStream("C:/temp/forest2.jpg");
BufferedInputStream bis = new BufferedInputStream(fis2);
start = System.currentTimeMillis();

while(bis.read() != -1) {}

end = System.currentTimeMillis();
System.out.println("사용했을 때: " + (end-start) + "ms");
bis.close();
fis2.close();
}
```

보조 스트림

❖ 성능 향상 보조 스트림

- BufferedInputStream, BufferedReader
 - 문자열 입력

```
Reader reader = new InputStreamReader(System.in);  
BufferedReader br = new BufferedReader(reader);  
String inputStr = br.readLine();
```

❖ 콘솔로부터 라인 단위로 읽기: `BufferedReaderExample.java`

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.Reader;

public class BufferedReaderExample {
    public static void main(String[] args) throws Exception {
        InputStream is = System.in;
        Reader reader = new InputStreamReader(is);
        BufferedReader br = new BufferedReader(reader);

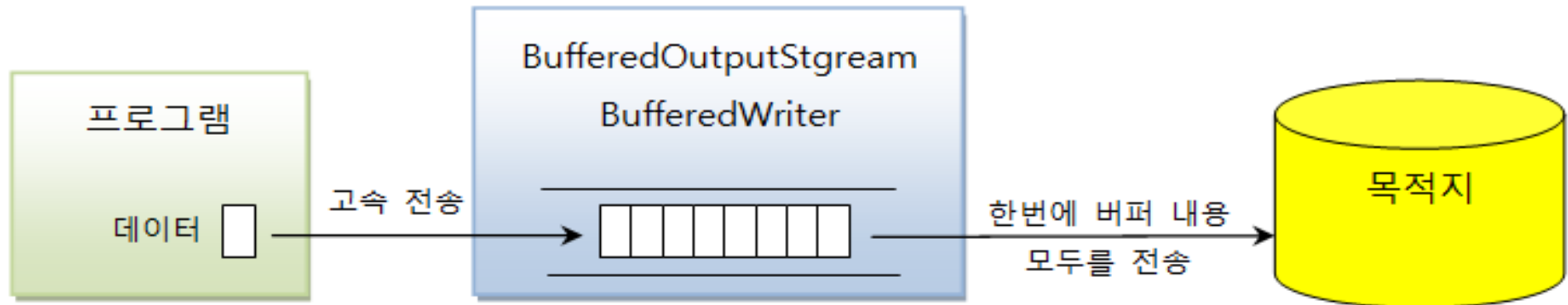
        System.out.print("입력: ");
        String lineString = br.readLine();

        System.out.println("출력: " + lineString);
    }
}
```

보조 스트림

❖ 성능 향상 보조 스트림

- BufferedOutputStream과 BufferedWriter



```
BufferedOutputStream bos = new BufferedOutputStream(바이트출력스트림);  
BufferedWriter bw = new BufferedWriter(문자출력스트림);
```

❖ 버퍼를 사용했을 때의 성능 테스트: BufferedOutputStreamExample.java

```
import java.io.BufferedInputStream;
import java.io.BufferedOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class BufferedOutputStreamExample {
    public static void main(String[] args) throws Exception {
        FileInputStream fis = null;
        FileOutputStream fos = null;
        BufferedInputStream bis = null;
        BufferedOutputStream bos = null;

        int data = -1;
        long start = 0;
        long end = 0;

        fis = new FileInputStream("C:/temp/forest.jpg");
        bis = new BufferedInputStream(fis);
        fos = new FileOutputStream("C:/Temp/forest_copy.jpg");
```


❖ 버퍼를 사용했을 때의 성능 테스트: BufferedOutputStreamExample.java

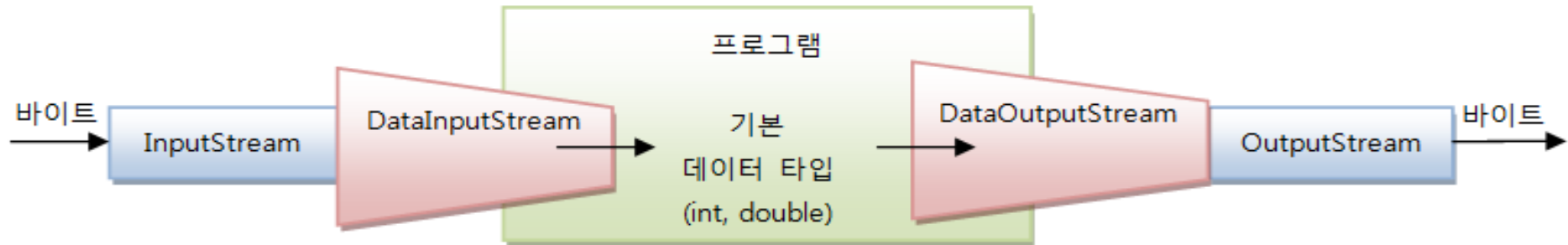
```
start = System.currentTimeMillis();
while((data = bis.read()) != -1) {
    fos.write(data);
}
fos.flush();
end = System.currentTimeMillis();
fos.close();    bis.close();    fis.close();
System.out.println("사용하지 않았을 때: " + (end-start) + "ms");

fis = new FileInputStream("C:/temp/forest.jpg");
bis = new BufferedInputStream(fis);
fos = new FileOutputStream("C:/Temp/forest.jpg");
bos = new BufferedOutputStream(fos);
start = System.currentTimeMillis();
while((data = bis.read()) != -1) {
    bos.write(data);
}
bos.flush();
end = System.currentTimeMillis();
bos.close(); fos.close();    bis.close(); fis.close();
System.out.println("사용했을 때: " + (end-start) + "ms");
}
```

보조 스트림

❖ 기본 타입 입출력 보조 스트림

- 입출력 순서를 맞추어 사용



```
DataInputStream dis = new DataInputStream(바이트입력스트림);  
DataOutputStream dos = new DataOutputStream(바이트출력스트림);
```

보조 스트림

❖ 기본 타입 입출력 보조 스트림

- 주요 메서드

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	writeInt(int v)
long	readLong()	void	writeLong(long v)
short	readShort()	void	writeShort(int v)
String	readUTF()	void	writeUTF(String str)

❖ 기본 데이터 타입 입출력: DataInputStreamExample.java

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.io.FileInputStream;
import java.io.FileOutputStream;

public class DataInputStreamExample {
    public static void main(String[] args) throws Exception {
        FileOutputStream fos = new FileOutputStream("C:/Temp/primitive.dat");
        DataOutputStream dos = new DataOutputStream(fos);

        dos.writeUTF("홍길동");
        dos.writeDouble(95.5);
        dos.writeInt(1);

        dos.writeUTF("감자바");
        dos.writeDouble(90.3);
        dos.writeInt(2);

        dos.flush(); dos.close(); fos.close();
    }
}
```

❖ 기본 데이터 타입 입출력: DataInputStreamExample.java

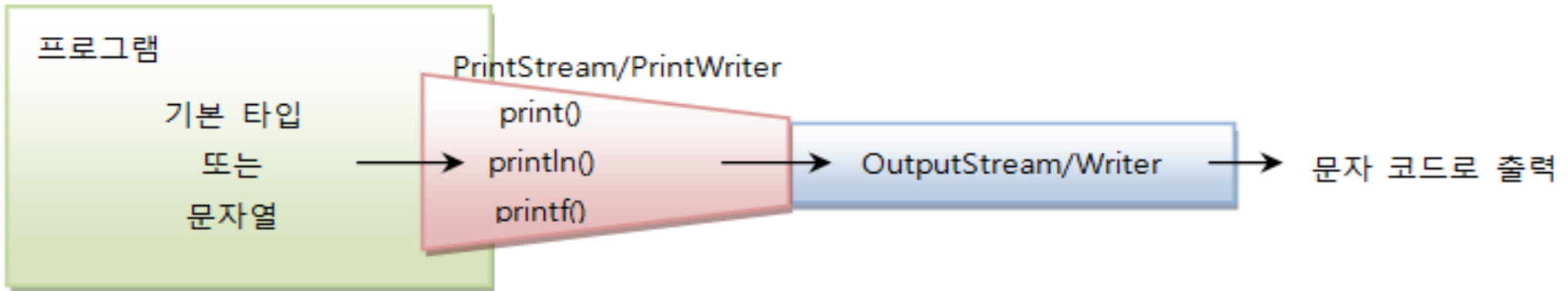
```
FileInputStream fis = new FileInputStream("C:/Temp/primitive.dat");
DataInputStream dis = new DataInputStream(fis);

for(int i=0; i<2; i++) {
    String name = dis.readUTF();
    double score = dis.readDouble();
    int order = dis.readInt();
    System.out.println(name + " : " + score + " : " + order);
}

dis.close(); fis.close();
}
```

보조 스트림

❖ 프린터 보조 스트림



```
PrintStream ps = new PrintStream(바이트출력스트림);  
PrintWriter pw = new PrintWriter(문자출력스트림);
```

보조 스트림

❖ 프린터 보조 스트림

PrintStream / PrintWriter

void	print(boolean b)
void	print(char c)
void	print(double d)
void	print(float f)
void	print(int i)
void	print(long l)
void	print(Object obj)
void	print(String s)

void	println(boolean b)
void	println(char c)
void	println(double d)
void	println(float f)
void	println(int i)
void	println(long l)
void	println(Object obj)
void	println(String s)
void	println()

- println()은 데이터 끝에 개행문자 추가, printf는 format string 출력

❖ 라인 단위로 출력하기: PrintStreamExample.java

```
import java.io.FileOutputStream;
import java.io.PrintStream;

public class PrintStreamExample {
    public static void main(String[] args) throws Exception {
        FileOutputStream fos = new FileOutputStream("C:/Temp/file.txt");
        PrintStream ps = new PrintStream(fos);

        ps.println("[프린터 보조 스트림]");
        ps.print("마치 ");
        ps.println("프린터가 출력하는 것처럼 ");
        ps.println("데이터를 출력합니다.");

        ps.flush();
        ps.close();
        fos.close();
    }
}
```


보조 스트림

❖ 프린터 보조 스트림

`printf(String format, Object... args)`

형식화된 문자열

형식화된 문자열에
제공될 매개값



`%` [argument_index\$] [flags] [width] [.precision] conversion

매개값의 순번

-, 0

전체자릿수

소수자릿수

변환문자

보조 스트림

❖ 프린터 보조 스트림

형식화된 문자		설명	출력 형태
정수	%d	정수	123
	%6d	6자리 정수. 왼쪽 빈자리 공백	__ 123
	%-6d	6자리 정수. 오른쪽 빈자리 공백	123__
	%06d	6자리 정수. 왼쪽 빈자리 0 채움	000123
실수	%10.2f	소수점 이상 7자리, 소수점 이하 2자리. 왼쪽 빈자리 공백	__ 123.45
	%-10.2f	소수점 이상 7자리, 소수점 이하 2자리. 오른쪽 빈자리 공백	123.45__
	%010.2f	소수점 이상 7자리, 소수점 이하 2자리. 왼쪽 빈자리 0 채움	0000123.45
문자열	%s	문자열	abc
	%6s	6자리 문자열. 왼쪽 빈자리 공백	__abc
	%-6s	6자리 문자열. 오른쪽 빈자리 공백	abc__

보조 스트림

❖ 프린터 보조 스트림

형식화된 문자		설명	출력 형태
날짜	%tY	4자리 년	2010
	%ty	2자리 년	10
	%tm	2자리 월	01
	%td	2자리 일	06
	%tH	2자리 시(0~23)	08
	%tI	시(0~12)	8
	%tM	2자리 분	06
	%tS	2자리 초	24
특수 문자	/t	탭(tab)	
	/n	줄바꿈	
	%%	%	%

❖ printf() 메서드 사용 방법: PrintfExample.java

```
import java.util.Date;

public class PrintfExample {
    public static void main(String[] args) {
        System.out.printf("상품의 가격:%d원\n", 123);
        System.out.printf("상품의 가격:%6d원\n", 123);
        System.out.printf("상품의 가격:%-6d원\n", 123);
        System.out.printf("상품의 가격:%06d원\n", 123);

        System.out.printf("반지름이 %d인 원의 넓이:%10.2f\n", 10, Math.PI*10*10);

        System.out.printf("%6d | %-10s | %10s\n", 1, "홍길동", "도적");

        Date now = new Date();
        System.out.printf("오늘은 %tY년 %tm월 %td일 입니다\n", now, now, now);
        System.out.printf("오늘은 %1$tY년 %1$tm월 %1$td일 입니다\n", now);
        System.out.printf("현재 %1$tH시 %1$tM분 %1$tS초 입니다\n", now);
    }
}
```