파일과 I/O 스트림

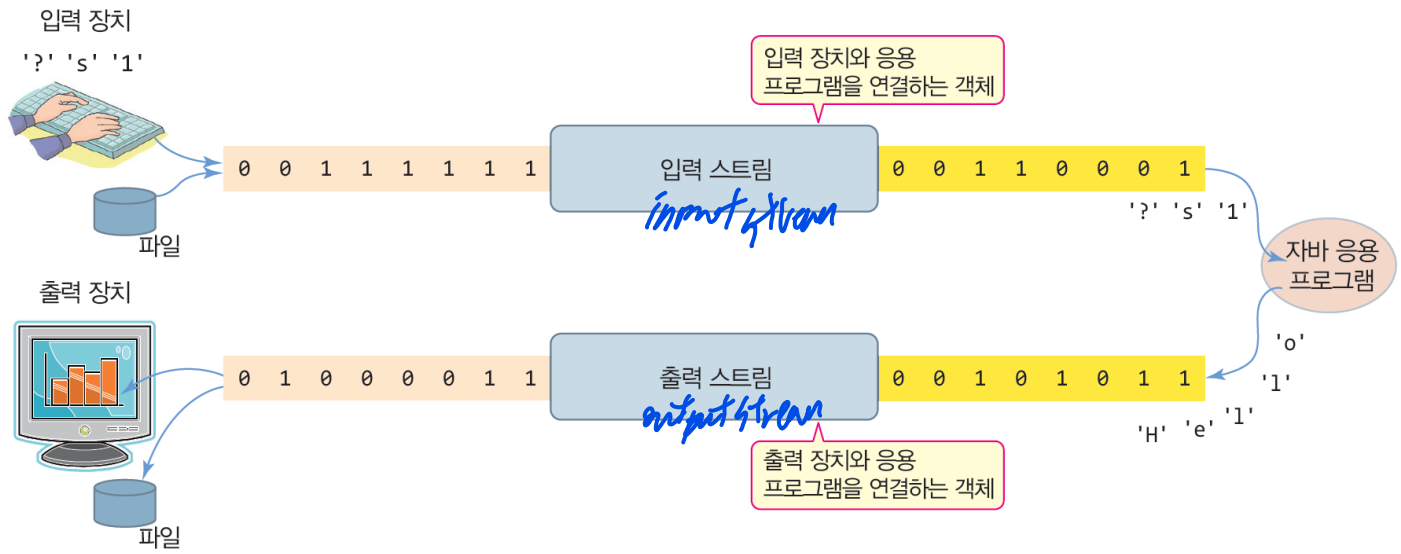
컴퓨터공학전공
박요한

수업내용

- I/O 소개
- 스트림의 이해
  - ✓문자 스트림
  - ✓바이트 스트림
- 파일 입출력
- 보조 스트림

스트림

- 스트림 입출력
  - ✓버퍼를 가지고 순차적으로 이루어지는 입출력
- 자바의 입출력 스트림
  - ✓응용프로그램과 입출력 장치를 연결하는 소프트웨어 모듈
    - 입력 스트림 : 입력 장치로부터 자바 프로그램으로 데이터를 전달
    - 출력 스트림 : 출력 장치로 데이터 출력



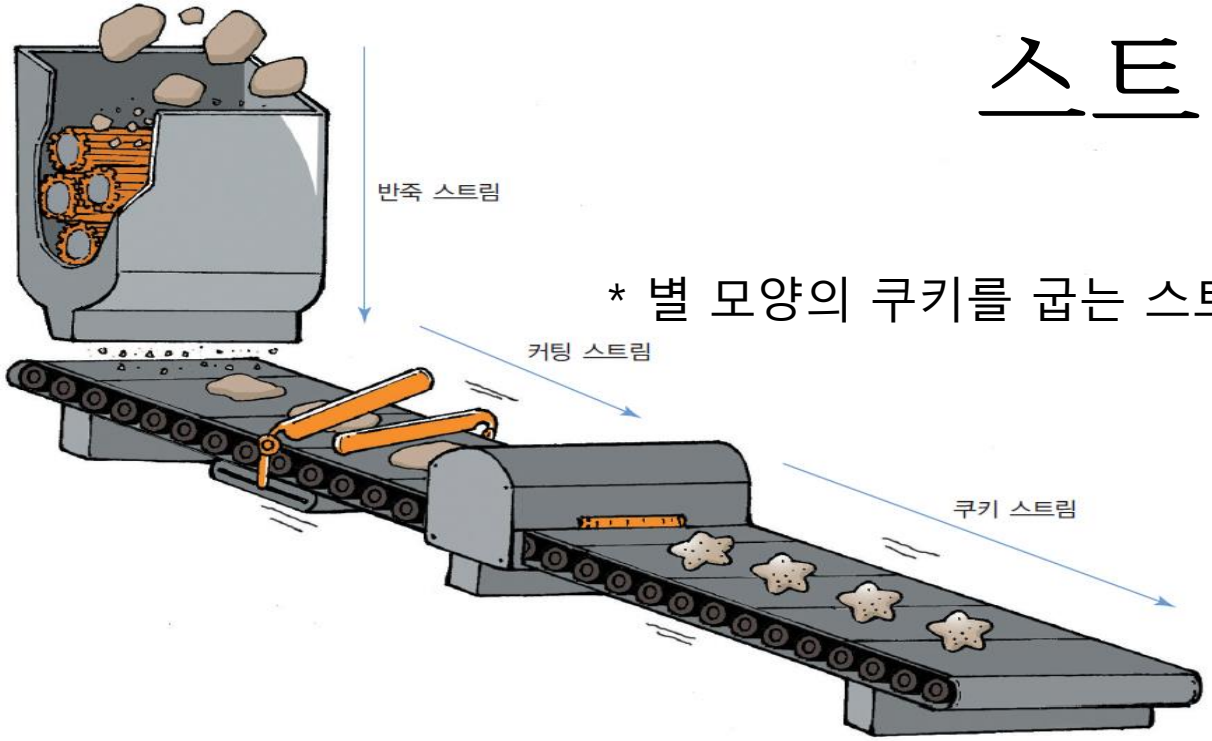
자바의 입출력 스트림 특징

- 스트림의 양끝에 입출력장치와 자바 응용프로그램 연결
- 스트림은 단방향
  - ✓입력과 출력을 동시에 하는 스트림 없음
- 입출력 스트림 기본 단위
  - ✓바이트 스트림의 경우 : 1 바이트
  - ✓문자 스트림의 경우 : 문자(자바에서는 문자1개 = 2 바이트)
- 선입선출(FIFO) 구조

자바의 입출력 스트림 종류

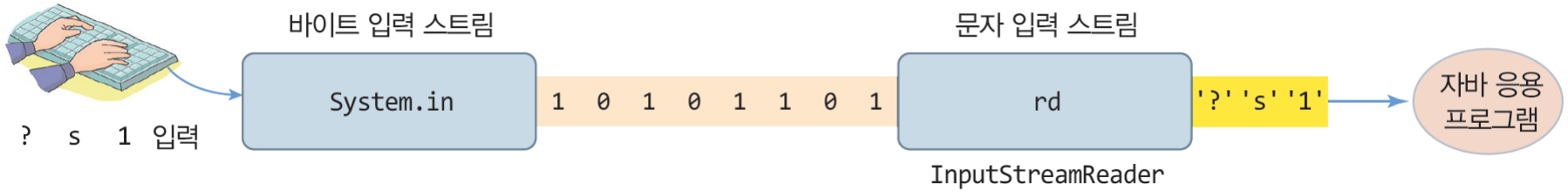
- 바이트 스트림과 문자 스트림
  - ✓바이트 스트림
    - 입출력되는 데이터를 단순 바이트로 처리
      - 예) 바이너리 파일을 읽는 입력 스트림
  - ✓문자 스트림
    - 문자만 입출력하는 스트림
    - 문자가 아닌 바이너리 데이터는 스트림에서 처리하지 못함
      - 예) 텍스트 파일을 읽는 입력 스트림
- JDK는 입출력 스트림을 구현한 다양한 클래스 제공

스트림은 연결될 수 있다



* 별 모양의 쿠키를 굽는 스트림

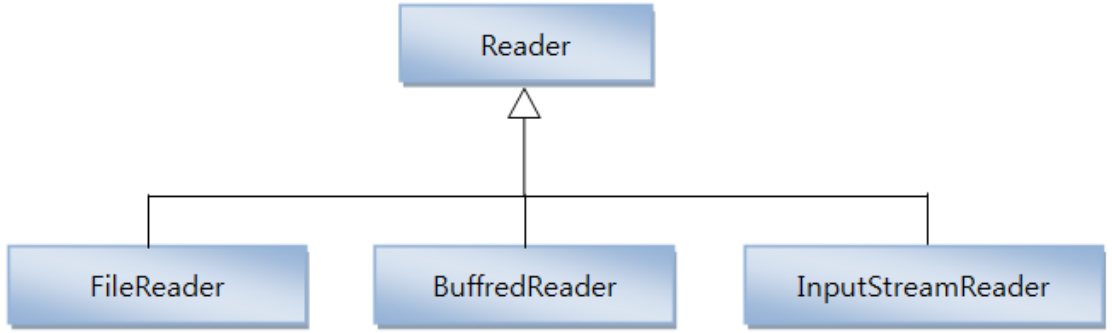
* 표준 입력 스트림 System.in에 InputStreamReader 스트림을 연결한 사례




```
InputStreamReader rd = new InputStreamReader(System.in);
int c = rd.read(); // 키보드에서 문자 읽음
```

입력 스트림과 출력 스트림

- Reader
  - ✓문자 기반 입력 스트림의 최상위 클래스로 추상 클래스

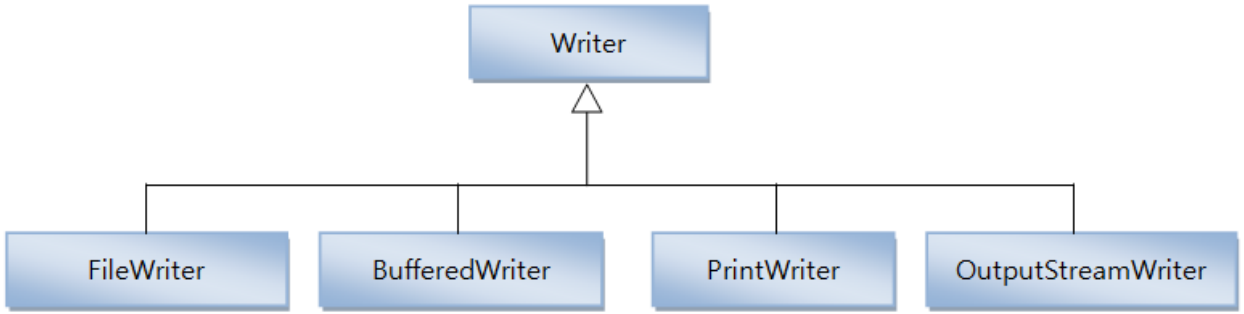


  - ✓Reader의 주요 메소드

메소드		설명
int	read()	입력 스트림으로부터 한개의 문자를 읽고 리턴한다.
int	read(char[] cbuf)	입력 스트림으로부터 읽은 문자들을 매개값으로 주어진 문자 배열 cbuf 에 저장하고 실제로 읽은 문자 수를 리턴한다.
int	read(char[] cbuf, int off, int len)	입력 스트림으로부터 len 개의 문자를 읽고 매개값으로 주어진 문자 배열 cbuf[off] 부터 len 개까지 저장한다. 그리고 실제로 읽은 문자 수인 len 개를 리턴한다.
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

입력 스트림과 출력 스트림

- Writer
  - ✓문자 기반 출력 스트림의 최상위 클래스로 추상 클래스



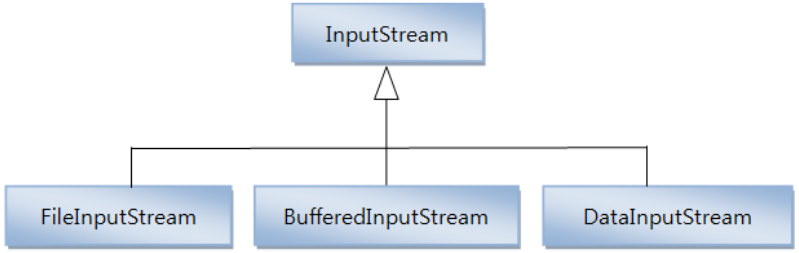
  - ✓Writer의 주요 메소드

리턴타입	메소드	설명
void	wrtie(int c)	출력 스트림으로 매개값으로 주어진 한 문자를 보낸다.
void	write(char[] cbuf)	출력 스트림에 매개값으로 주어진 문자 배열 cbuf 의 모든 문자를 보낸다.
void	write(char[] cbuf, int off, int len)	출력 스트림에 매개값으로 주어진 문자 배열 cbuf[off] 부터 len 개까지의 문자를 보낸다.
void	write(String str)	출력 스트림에 매개값으로 주어진 문자열을 전부 보낸다.
void	write(String str, int off, int len)	출력 스트림에 매개값으로 주어진 문자열 off 순번부터 len 개까지의 문자를 보낸다.
void	flush()	버퍼에 잔류하는 모든 문자열을 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

파일 입출력
- 바이트 스트림

입력 스트림과 출력 스트림

- InputStream
  - ✓바이트 기반 입력 스트림의 최상위 클래스로 추상 클래스

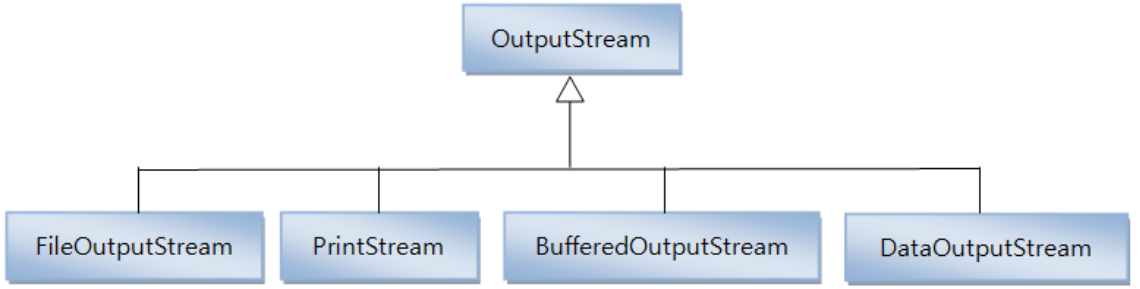


  - ✓InputStream 클래스의 주요 메소드

리턴타입	메소드	설명
int	read()	입력 스트림으로부터 1 바이트를 읽고 읽은 바이트를 리턴한다.
int	read(byte[] b)	입력 스트림으로부터 읽은 바이트들을 매개값으로 주어진 바이트 배열 b 에 저장하고 실제로 읽은 바이트 수를 리턴한다.
int	read(byte[] b, int off, int len)	입력 스트림으로부터 len 개의 바이트 만큼 읽고 매개값으로 주어진 바이트 배열 b[off] 부터 len 개까지 저장한다. 그리고 실제로 읽은 바이트 수인 len 개를 리턴한다. 만약 len 개를 모두 읽지 못하면 실제로 읽은 바이트 수를 리턴한다.
void	close()	사용한 시스템 자원을 반납하고 입력 스트림을 닫는다.

입력 스트림과 출력 스트림

- OutputStream
  - ✓바이트 기반 출력 스트림의 최상위 클래스로 추상 클래스



  - ✓OutputStream의 주요 메소드

리턴타입	메소드	설명
void	wrtie(int b)	출력 스트림으로 1 바이트를 보낸다.
void	write(byte[] b)	출력 스트림에 매개값으로 주어진 바이트 배열 b 의 모든 바이트를 보낸다.
void	write(byte[] b, int off, int len)	출력 스트림에 매개값으로 주어진 바이트 배열 b[off] 부터 len 개까지의 바이트를 보낸다.
void	flush()	버퍼에 잔류하는 모든 바이트를 출력한다.
void	close()	사용한 시스템 자원을 반납하고 출력 스트림을 닫는다.

바이트 스트림 클래스

- 바이트 스트림
  - ✓바이트 단위의 바이너리 값을 읽고 쓰는 스트림
- 바이트 스트림 클래스
  - ✓InputStream/OutputStream
    - 추상 클래스
    - 바이트 스트림을 다루는 모든 클래스의 슈퍼 클래스
  - ✓FileInputStream/FileOutputStream
    - 파일로부터 바이트 단위로 읽거나 저장하는 클래스
    - 바이너리 파일의 입출력 용도
  - ✓DataInputStream/DataOutputStream
    - 자바의 기본 데이터 타입의 값(변수)을 바이너리 값 그대로 입출력
    - 문자열도 바이너리 형태로 입출력

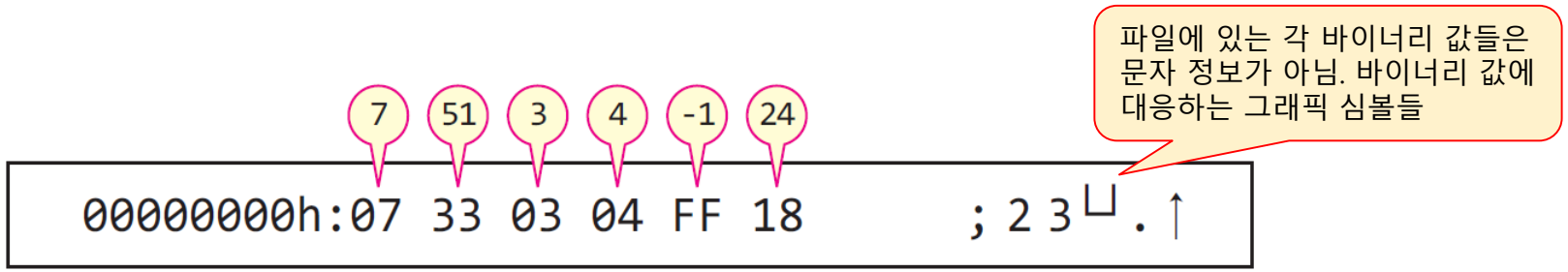
FileOutputStream을 이용한 파일 쓰기

- 바이너리 값을 파일에 저장하는 바이트 스트림 코드

```
FileOutputStream fout = new FileOutputStream("c:₩₩Temp₩₩test.out");

byte b[] = {7,51,3,4,-1,24};
for(int i=0; i<b.length; i++)
   fout.write(b[i]);

fout.close();
```

파일에 배열 b[i]의 정수 값(바이너리)을 그대로 기록

스트림을 닫음. 파일도 닫힘. 더 이상 스트림으로부터 읽을 수 없음



test.out 파일의 내부

보조 스트림

- 버퍼 스트림

버퍼 입출력 스트림과 버퍼 입출력의 특징

- 버퍼 스트림
  - ✓버퍼를 가진 스트림
  - ✓입출력 데이터를 일시적으로 저장하는 버퍼를 이용하여 입출력 효율 개선
- 버퍼 입출력의 목적
  - ✓입출력 시 운영체제의 API 호출 횟수를 줄여 입출력 성능 개선
    - 출력시 여러 번 출력되는 데이터를 버퍼에 모아두고 한 번에 장치로 출력
    - 입력시 입력 데이터를 버퍼에 모아두고 한번에 프로그램에게 전달

버퍼 스트림의 종류

- 바이트 버퍼 스트림
  - ✓바이트 단위의 바이너리 데이터를 처리하는 버퍼 스트림
  - ✓BufferedInputStream와 BufferedOutputStream
- 문자 버퍼 스트림
  - ✓유니코드의 문자 데이터만 처리하는 버퍼 스트림
  - ✓BufferedReader와 BufferedWriter

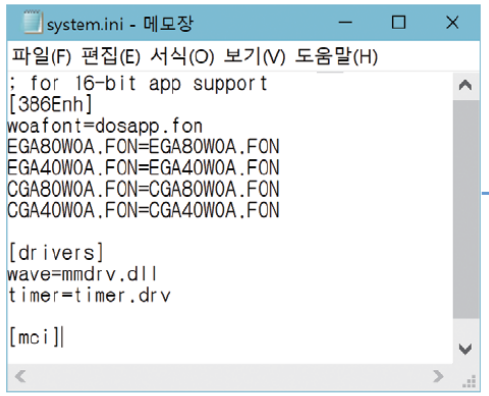
20바이트 버퍼를 가진 BufferedOutputStream

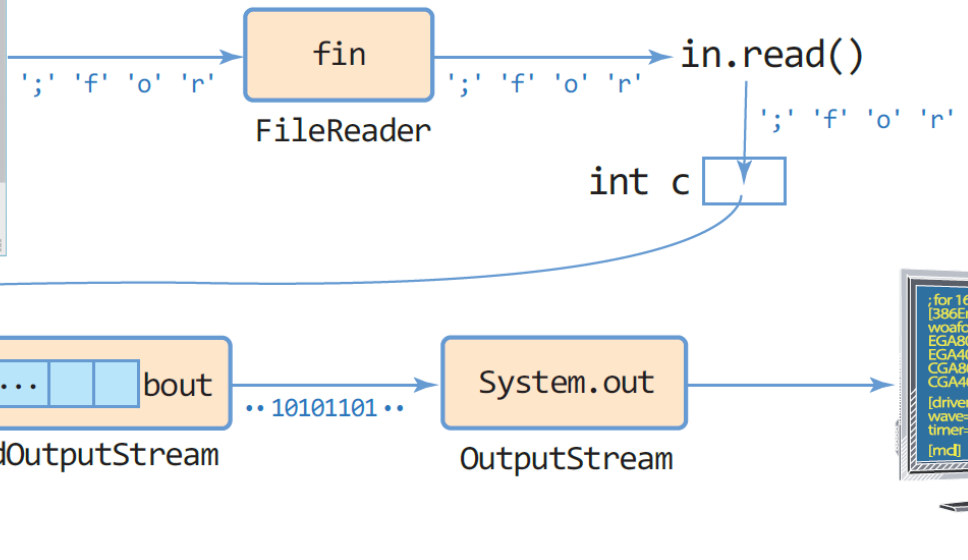
```
BufferedOutputStream bout = new BufferedOutputStream(System.out, 20);
FileReader fin = new FileReader("c:₩₩windows₩₩system.ini");

int c;
while ((c = fin.read()) != -1) {
   bout.write((char)c);
}
fin.close();
bout.close();
```

20바이트 크기의 버퍼 설정.
System.out 표준 스트림에 출력

파일 전체를 읽어 화면에 출력

스트림 닫음



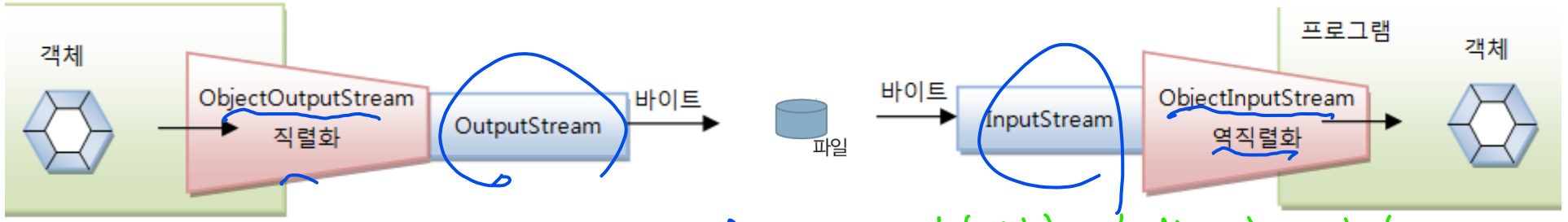


보조 스트림

- Object 스트림

객체 입출력 스트림

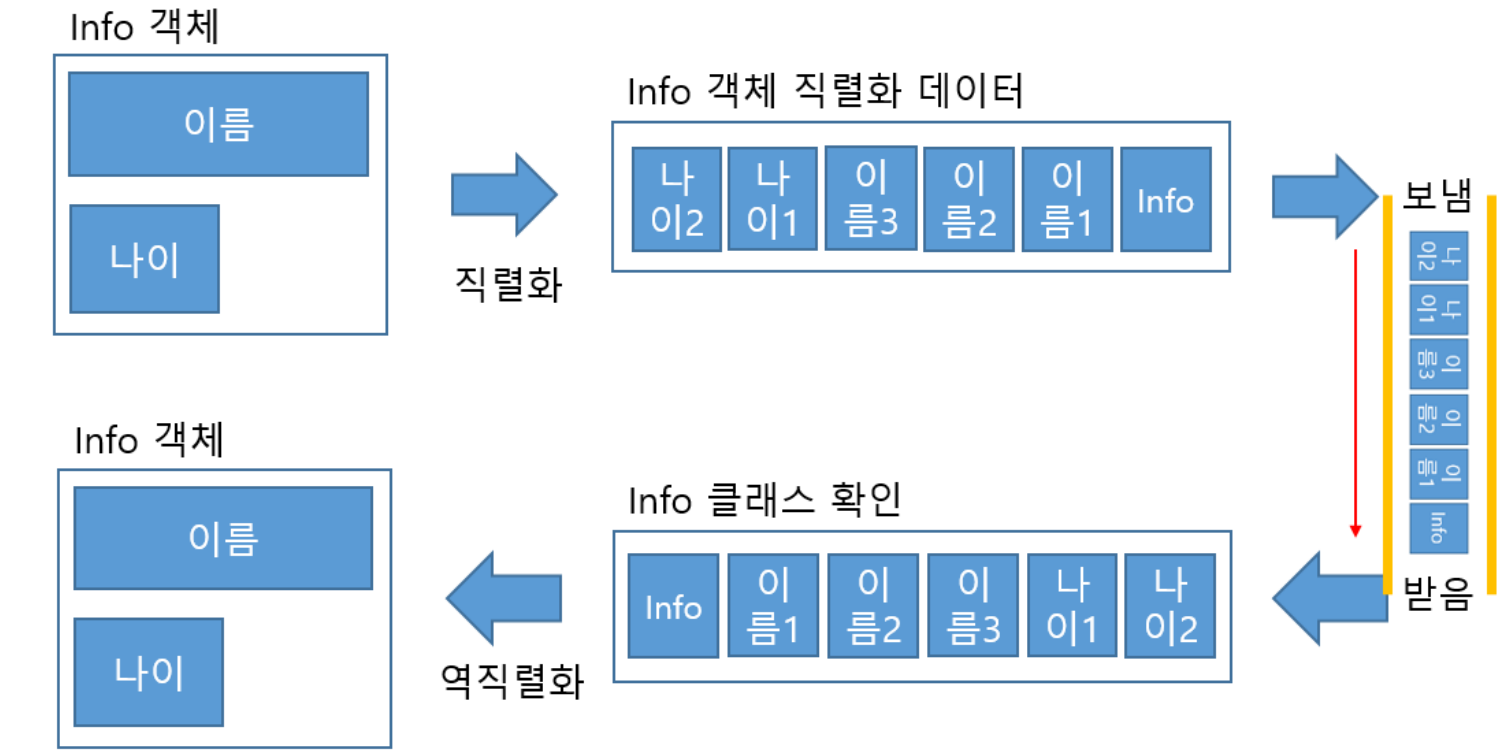
- 객체 입출력 보조 스트림
  - ✓객체를 파일로 입출력 할 수 있는 기능 제공
  - ✓객체 직렬화
    - 객체는 문자가 아니므로 바이트 기반 스트림으로 데이터 변경 필요
  - ✓ObjectInputStream, ObjectOutputStream



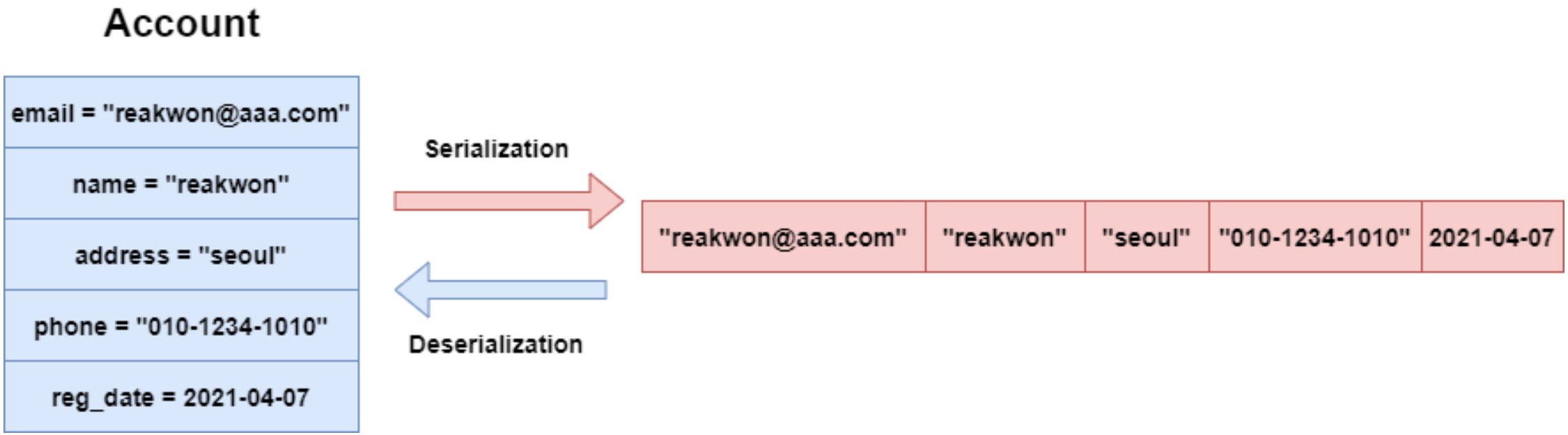
  - ✓직렬화가 가능한 클래스(Serializable)
    - 자바에서는 Serializable 인터페이스를 구현한 클래스만 직렬화 할 수 있도록 제한
    - 객체 직렬화 할 때 private 필드 포함한 모든 필드를 바이트로 변환 가능

직렬화(Serializable)

```
class Info{
        String name;
        int age;
}
```



직렬화(Serializable)



객체 입출력 스트림

✓ serialVersionUID 필드 (p.1047~1049)

- 직렬화된 객체를 역직렬화 할 때는 직렬화 했을 때와 같은 클래스 사용

- 클래스의 이름이 같더라도 클래스의 내용이 변경된 경우 역직렬화 실패

- serialVersionUID
  - 같은 클래스임을 알려주는 식별자 역할
  - Serializable 인터페이스 구현
    - 컴파일 시 자동적으로 serialVersionUID 정적 필드 추가
  - 재컴파일하면 serialVersionUID의 값 변경

- 불가피한 수정 있을 경우 명시적으로 serialVersionUID 선언

```
static final long serialVersionUID = 정수값;
```

객체 입출력 스트림

- ✓ writeObject()와 readObject() 메소드
  - writeObject(ObjectOutputStream out)
  - 직렬화 직전 자동 호출
    - 추가 직렬화할 내용 작성 가능

  - readObject(ObjectInputStream in)
  - 역직렬화 직전 자동 호출
    - 추가 역직렬화 내용 작성 가능