소켓(Socket)

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓에 대하여

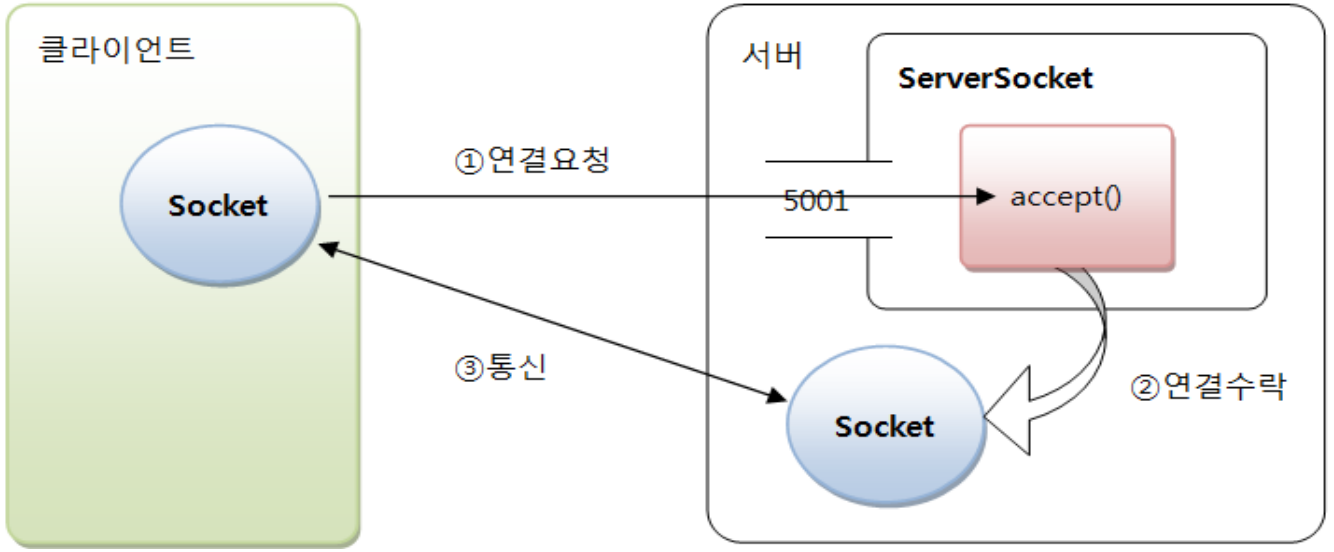
- 소켓(socket) : 프로그램 내에서 보았을 때의 데이터 통신 출입구
  - java.net API
  - 서버 소켓, 클라이언트 소켓 두 종류
    - ServerSocket, Socket

- ServerSocket
  - 서버 프로그램에서만 사용되는 소켓
  - 연결 요청을 기다리다가, 연결 요청이 오면 연결을 맺고 또 다른 소켓을 생성

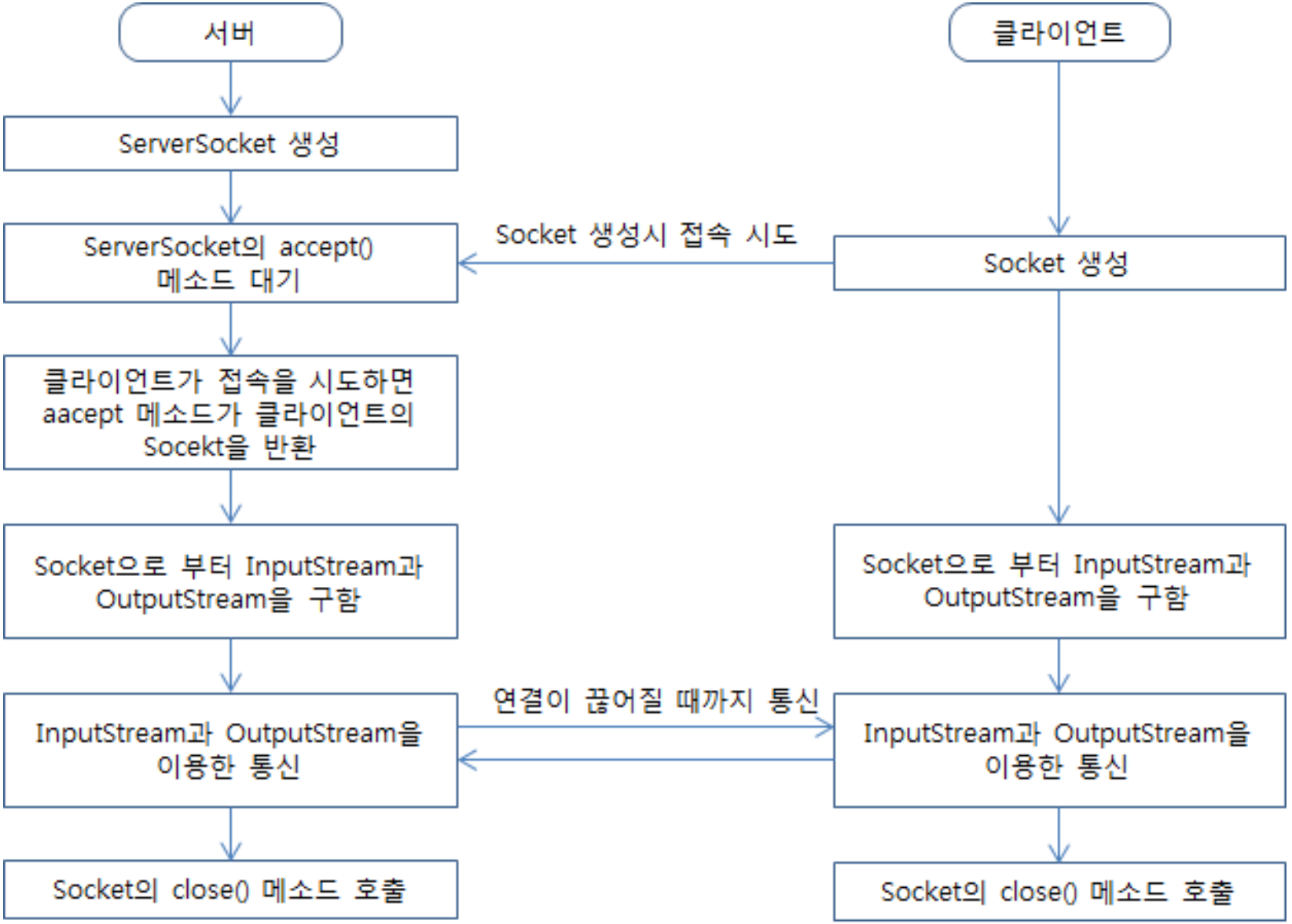
- Socket
  - 클라이언트 프로그램과 서버 프로그램에서 모두 사용되는 소켓
  - 실제 데이터 전송에 사용되는 것은 이 소켓임
  - 서버 프로그램에서는 서버 소켓에 의해 생성됨
  - 클라이언트 프로그램에서는 직접 생성해야 함

TCP/IP 통신 프로그램의 작성 방법

❖ ServerSocket과 Socket 용도



TCP/IP 통신 프로그램의 작성 방법



TCP/IP 통신 프로그램의 작성 방법

❖ ServerSocket 생성과 연결 수락

- ServerSocket 생성과 포트 바인딩
  - 생성자에 바인딩 포트 대입하고 객체 생성

- 연결 수락
  - accept() 메소드는 클라이언트가 연결 요청 전까지 블로킹 대기
  - 연결된 클라이언트 IP 주소 얻기

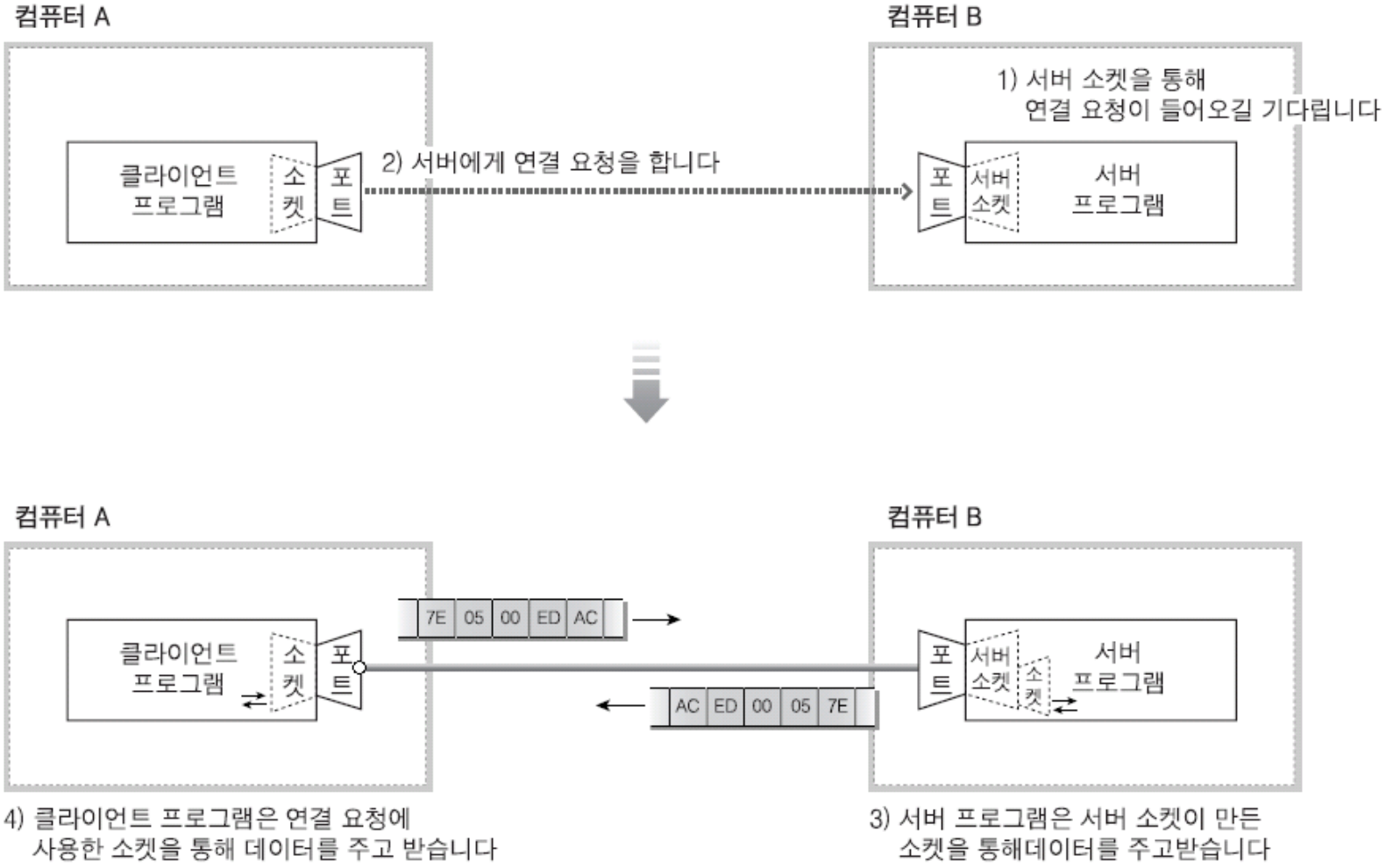
```
InetSocketAddress socketAddress = (InetSocketAddress) socket.getRemoteSocketAddress();
```

리터타입	메소드명(매개변수)	설명
String	getHostName()	클라이언트 IP 리턴
int	getPort()	클라이언트 포트 번호 리턴
String	toString()	"IP:포트번호" 형태의 문자열 리턴

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신

- ㅇ 소켓을 이용한 클라이언트 프로그램과 서버 프로그램의 통신 과정



TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 서버 프로그램

- ㅇ 서버 소켓을 생성하고 사용하는 방법
  1) ServerSocket 객체를 생성합니다.

```
ServerSocket serverSocket = new ServerSocket(9000);
```

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 서버 프로그램

- ㅇ 서버 소켓을 생성하고 사용하는 방법

  2) ServerSocket 객체에 대해 accept 메소드를 호출합니다.

```
Socket socket = serverSocket.accept();
```

서버 소켓으로 연결 요청이 들어오면 연결을 맺고,
클라이언트 소켓을 생성해서 리턴하는 메소드

TCP/IP 통신 프로그램의 작성 방법

❖ **소켓을 이용한 통신 : 서버 프로그램**

ㅇ 서버 소켓을 모두 사용하고 난 후에는 닫아야 합니다.

```
serverSocket.close();
```

서버 소켓을 닫는 메소드

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 서버 프로그램

- ㅇ 기본 골격
  - ▪ try-with-resource()를 사용하는 경우 서버 소켓을 자동으로 close()

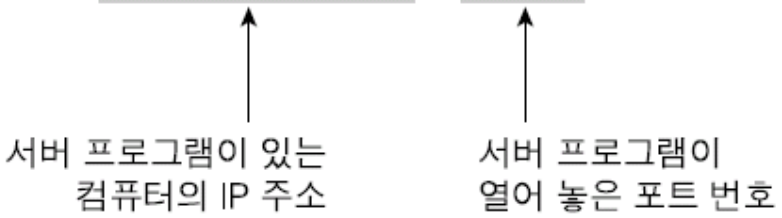
```
try(ServerSocket server = new ServerSocket(포트_번호)) {
   Socket socket = server.accept();    // 접속 대기

      // 클라이언트 소켓(Socket)으로 통신 수행     :

   socket.close();
} catch (Exception e) {
   e.printStackTrace();
}
```

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 클라이언트/서버 프로그램

- ㅇ 클라이언트 소켓을 생성하고 사용하는 방법
  1) Socket 객체를 생성합니다.

```
Socket socket = new Socket("219.153.21.14", 9000);
```



TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 클라이언트/서버 프로그램

- ㅇ 클라이언트 소켓을 생성하고 사용하는 방법

2) 데이터 송수신에 사용할 입력 스트림 객체와 출력 스트림 객체를 얻습니다.

```
InputStream in = socket.getInputStream();
```

데이터 수신에 사용할
입력 스트림 객체를 리턴하는 메소드

```
OutputStream out = socket.getOutputStream();
```

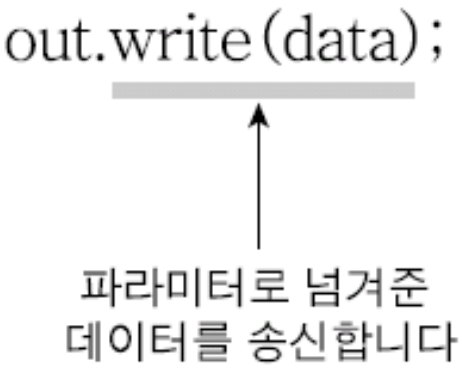
데이터 송신에 사용할
출력 스트림 객체를 리턴하는 메소드

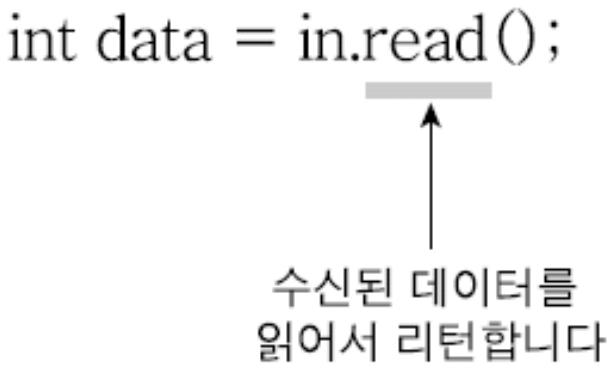
TCP/IP 통신 프로그램의 작성 방법

❖ **소켓을 이용한 통신 : 클라이언트/서버 프로그램**

- 클라이언트 소켓을 생성하고 사용하는 방법

  3) write 메소드와 read 메소드를 호출하여 데이터 송신 또는 수신합니다.

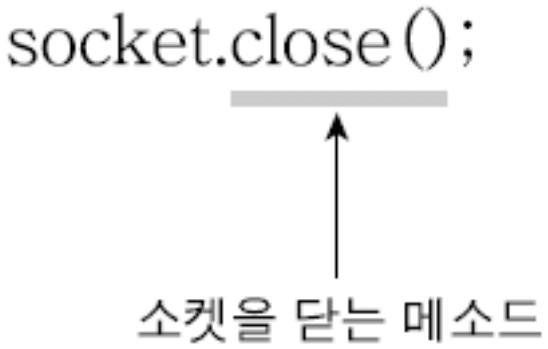




❖ **소켓을 이용한 통신 : 클라이언트/서버 프로그램**

ㅇ 클라이언트 소켓을 생성하고 사용하는 방법

4) 클라이언트 소켓을 닫습니다.



TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 서버 프로그램

- 기본 골격

```
try(ServerSocket server = new ServerSocket(포트_번호)) {
    Socket socket = server.accept();    // 접속 대기
    // 수신 Reader 준비
    BufferedReader r = new BufferedReader(
            new InputStreamReader(socket.getInputStream()));
    // 전송 Writer 준비
    PrintWriter w = new PrintWriter(
            socket.getOutputStream());

    // 메시지 수신

    // 메시지 전송
    w.flush();

    socket.close();

} catch (Exception e) {
    e.printStackTrace();
}
```

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 클라이언트 프로그램

- ㅇ 기본 골격
  - ▪ 서버 연결 – Socket 생성 성공은 서버와의 연결됬음을 의미

```
try(Socket socket = new Socket("127.0.0.1", 10000)) {

      // 스트림을 통한 통신 수행

} catch (Exception e) {
   e.printStackTrace();
}
```

TCP/IP 통신 프로그램의 작성 방법

❖ 소켓을 이용한 통신 : 클라이언트 프로그램

- 기본 골격 : 메시지 수신 및 전송을 위한 스트림 추출 후 작업 수행

```
try(Socket socket = new Socket("127.0.0.1", 10000)) {
    InputStream is = socket.getInputStream();
    OutputStream out = socket.getOutputStream();

    // out으로 메시지 전송
        :
    out.flush(); // 메시지를 실제로 전송

    // is으로 메시지 수신
        :

} catch (Exception e) {
    e.printStackTrace();
}
```