

# TCP/IP

# TCP/IP 프로토콜에 대하여

---

## ❖ TCP/IP 프로토콜

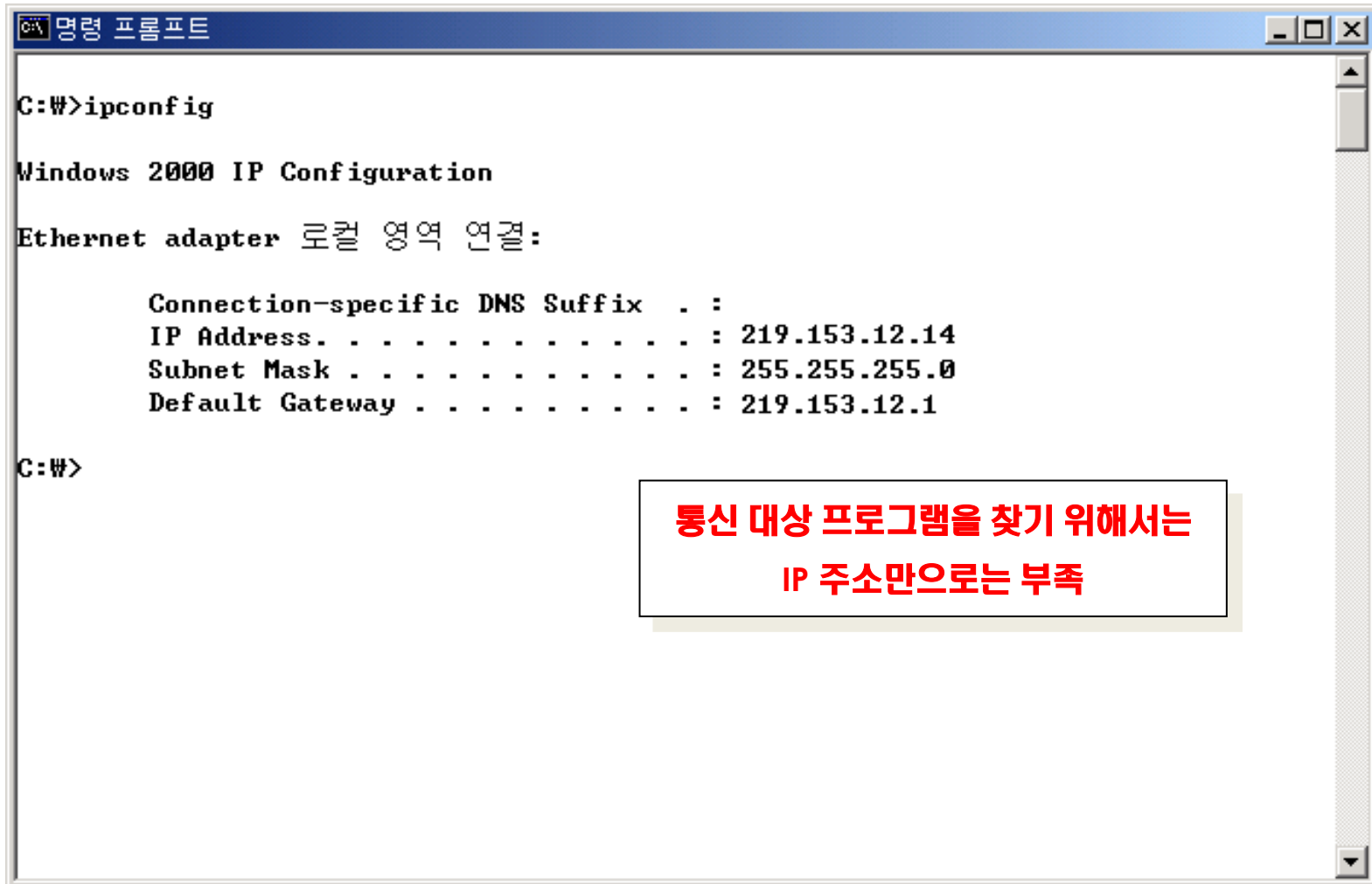
- 인터넷에서 사용되는 프로토콜(protocol, 통신 규칙)
- TCP 프로토콜과 IP 프로토콜을 함께 부르는 이름

↑  
**네트워크에 연결된 각각의 컴퓨터에  
IP 주소(IP address)를 붙여서 관리하는 규칙**

# TCP/IP 프로토콜에 대하여

## ❖ TCP/IP 프로토콜

- 컴퓨터의 IP 주소 확인하는 방법 : ipconfig.exe



```
명령 프롬프트

C:\>ipconfig

Windows 2000 IP Configuration

Ethernet adapter 로컬 영역 연결:

    Connection-specific DNS Suffix  . : 
    IP Address. . . . .               : 219.153.12.14
    Subnet Mask . . . . .             : 255.255.255.0
    Default Gateway . . . . .         : 219.153.12.1

C:\>
```

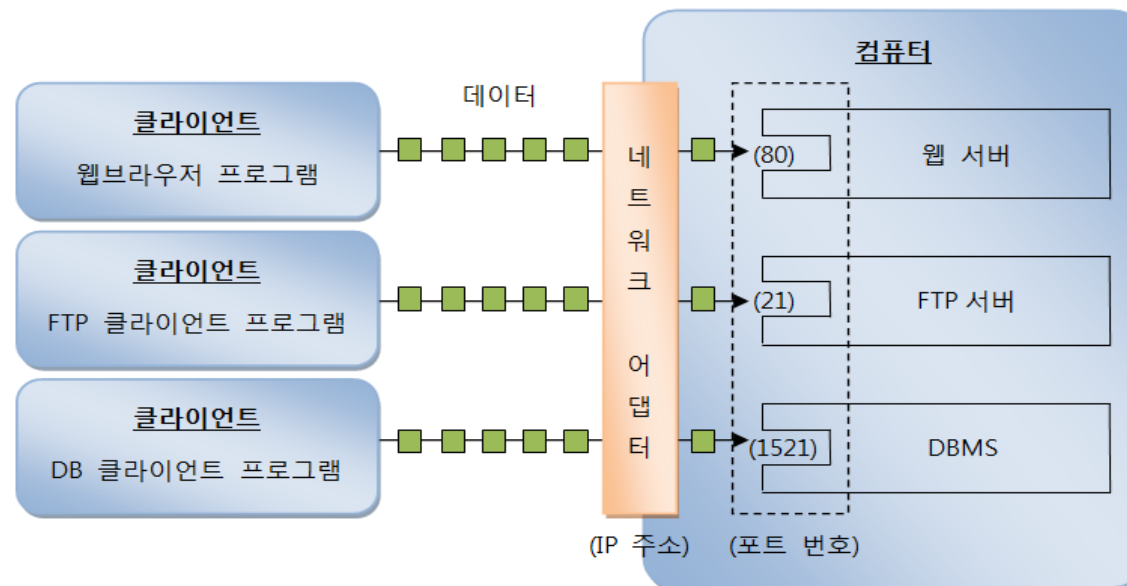
통신 대상 프로그램을 찾기 위해서는  
IP 주소만으로는 부족

# TCP/IP 프로토콜에 대하여

## ❖ TCP/IP 프로토콜

- 포트(port) : 네트워크를 통해 데이터를 주고 받을 때 사용되는 가상의 출구
  - 0 ~ 65535 범위의 정수 번호로 식별됨

구분명	범위	설명
Well Know Port Numbers	0~1023	국제인터넷주소관리기구(ICANN)가 특정 애플리케이션용으로 미리 예약한 포트
Registered Port Numbers	1024~49151	회사에서 등록해서 사용할 수 있는 포트
Dynamic Or Private Port Numbers	49152~65535	운영체제가 부여하는 동적 포트 또는 개인적인 목적으로 사용할 수 있는 포트



# TCP/IP 프로토콜에 대하여

## ❖ TCP/IP 프로토콜

- 컴퓨터에서 사용 중인 포트 번호 확인하는 방법 : netstat.exe

```
C:\>netstat -na

Active Connections

Proto Local Address          Foreign Address         State
TCP   0.0.0.0:135             0.0.0.0:0               LISTENING
TCP   0.0.0.0:445             0.0.0.0:0               LISTENING
UDP   0.0.0.0:445             *:*
```

사용 중인 포트 번호

주의: 0 ~ 1023까지는 시스템이 사용하는 포트번호

# TCP/IP 프로토콜에 대하여

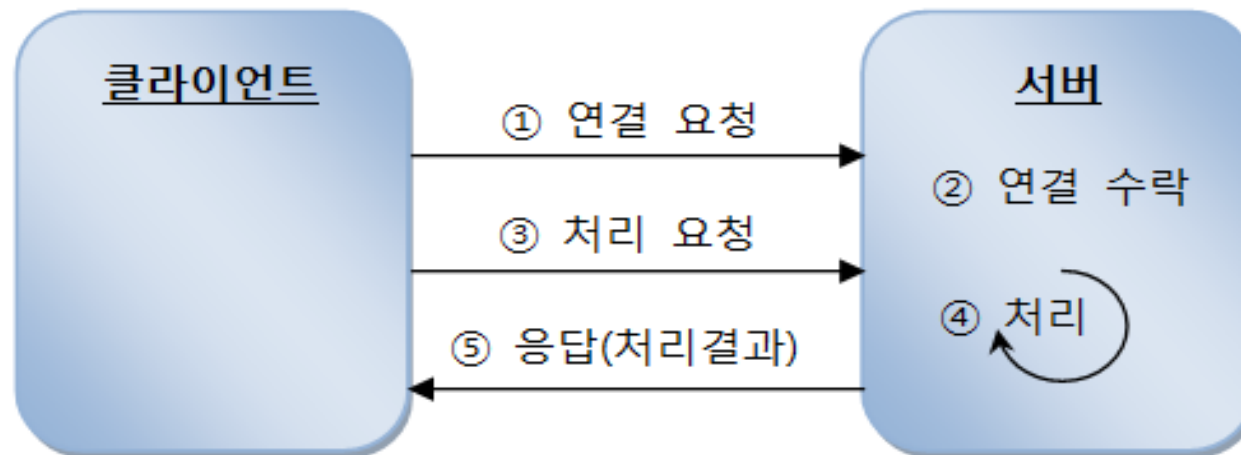
## ❖ 네트워크 통신 프로그램

### ○ 통신 프로그램의 작성 방법

- 1) 한 프로그램은 연결을 요청하고 다른 프로그램은 그 요청을 받아서 연결을 맺습니다.
- 2) 그 연결을 통해 데이터를 주고 받습니다.

### ○ 용어 설명

- 클라이언트 프로그램(client program) : 연결을 요청하는 통신 프로그램
- 서버 프로그램(server program) : 연결 요청을 기다리는 통신 프로그램



# TCP/IP 프로토콜에 대하여

---

## ❖ InetAddress로 IP 주소 얻기

- `java.net.InetAddress`
- IP 주소 표현한 클래스
- 로컬 컴퓨터의 IP 주소
- 도메인 이름을 DNS에서 검색한 후 IP 주소를 가져오는 기능 제공

# TCP/IP 프로토콜에 대하여

## ❖ InetAddress로 IP 주소 얻기

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class InetAddressExample {
    public static void main(String[] args) {
        try {
            InetAddress local = InetAddress.getLocalHost();
            System.out.println("내컴퓨터 IP주소: " + local.getHostAddress();

            InetAddress[] iaArr = InetAddress.getAllByName("www.naver.com");
            for(InetAddress remote : iaArr) {
                System.out.println("www.naver.com IP주소: " +
                                   remote.getHostAddress();
            }
        } catch(UnknownHostException e) {
            e.printStackTrace();
        }
    }
}
```



# TCP/IP 프로토콜에 대하여

---

## ❖ TCP(Transmission Control Protocol)

### ○ 특징

- 연결 지향적 프로토콜 -> 시간 소요
- 통신 선로 고정 -> 전송 속도 느려질 수 있음
- 데이터를 정확하고 안정적으로 전달

# TCP/IP 통신 프로그램의 작성 방법

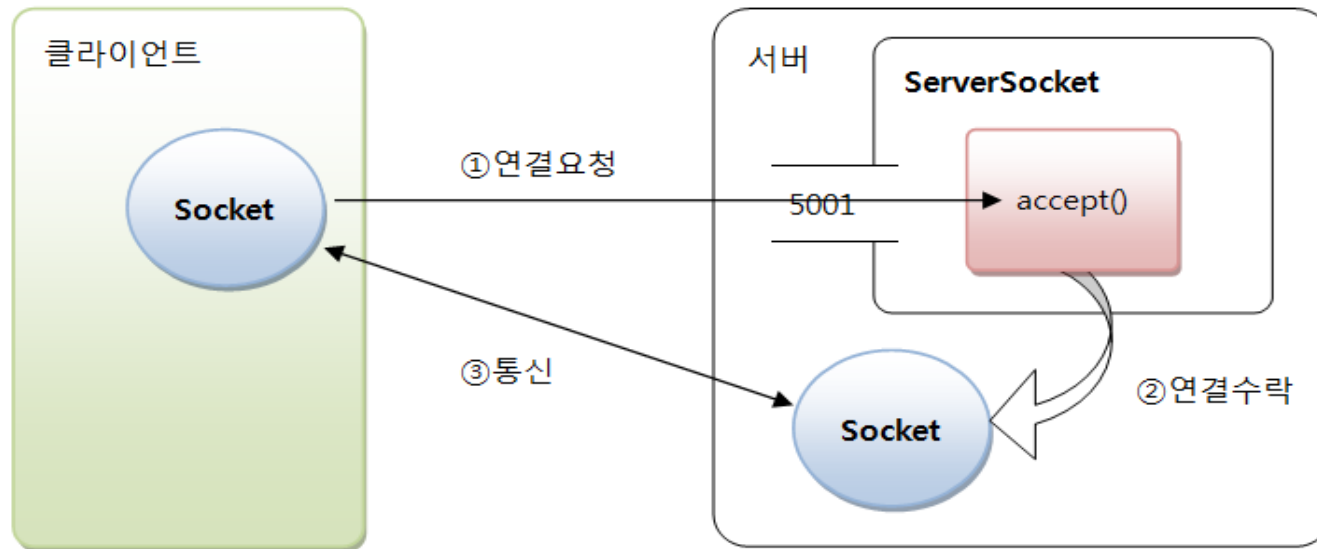
---

## ❖ 소켓에 대하여

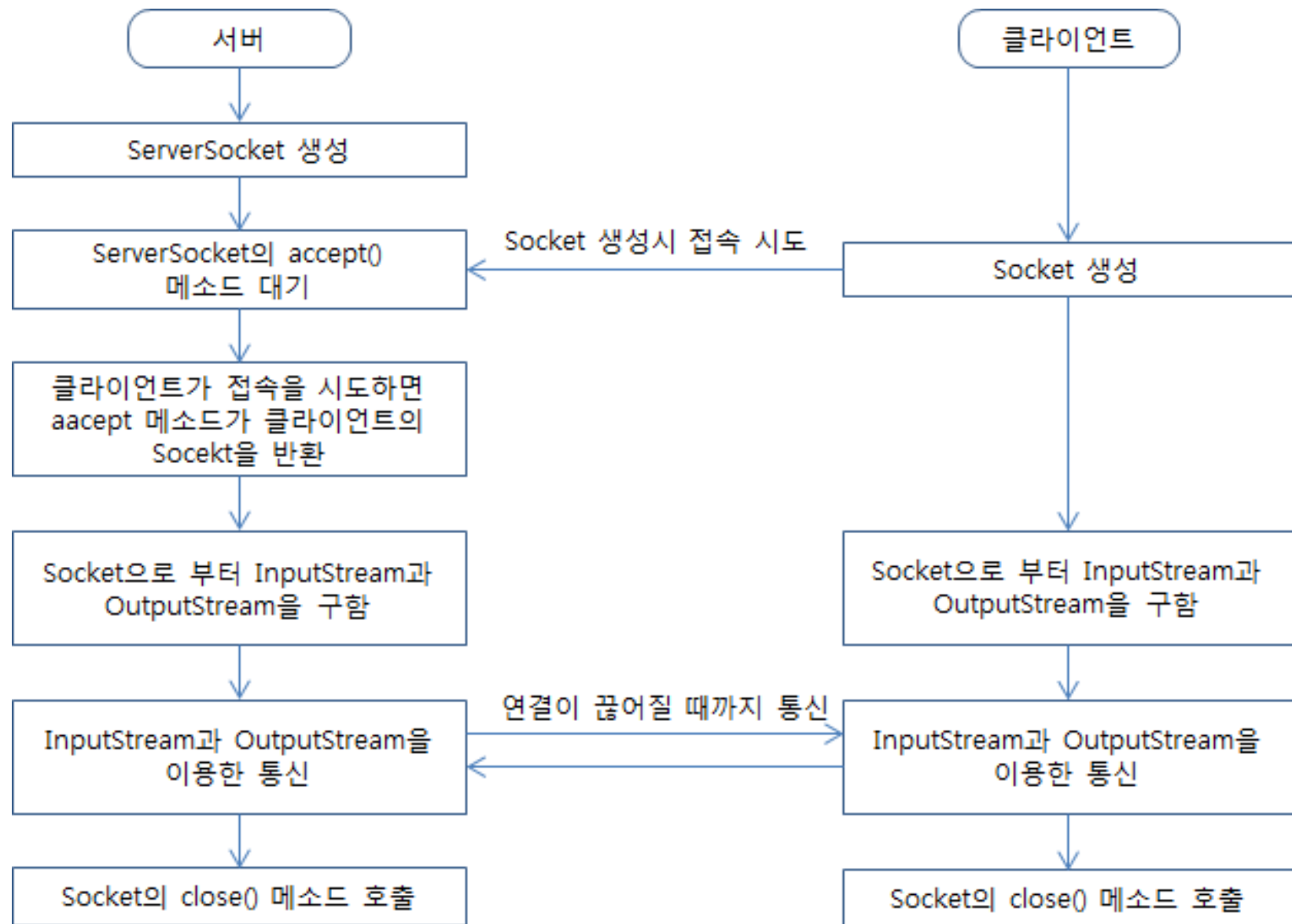
- 소켓(socket) : 프로그램 내에서 보았을 때의 데이터 통신 출입구
  - java.net API
  - 서버 소켓, 클라이언트 소켓 두 종류
    - ServerSocket, Socket
- ServerSocket
  - 서버 프로그램에서만 사용되는 소켓
  - 연결 요청을 기다리다가, 연결 요청이 오면 연결을 맺고 또 다른 소켓을 생성
- Socket
  - 클라이언트 프로그램과 서버 프로그램에서 모두 사용되는 소켓
  - 실제 데이터 전송에 사용되는 것은 이 소켓임
  - 서버 프로그램에서는 서버 소켓에 의해 생성됨
  - 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();
```

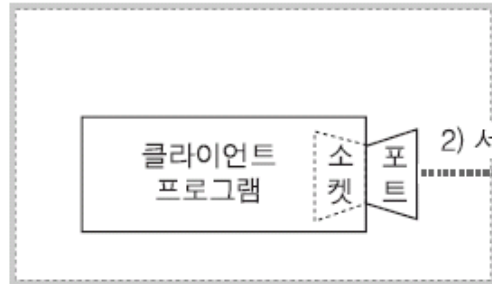
리터타입	메소드명(매개변수)	설명
String	<code>getHostName()</code>	클라이언트 IP 리턴
int	<code>getPort()</code>	클라이언트 포트 번호 리턴
String	<code>toString()</code>	"IP:포트번호" 형태의 문자열 리턴

# TCP/IP 통신 프로그램의 작성 방법

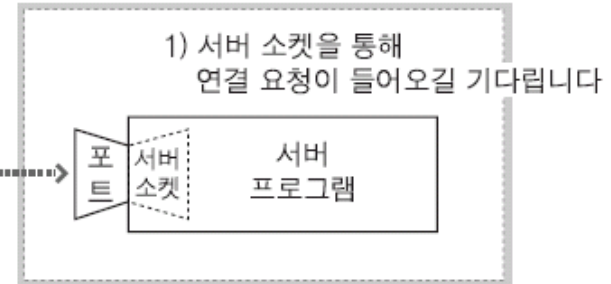
## ❖ 소켓을 이용한 통신

- 소켓을 이용한 클라이언트 프로그램과 서버 프로그램의 통신 과정

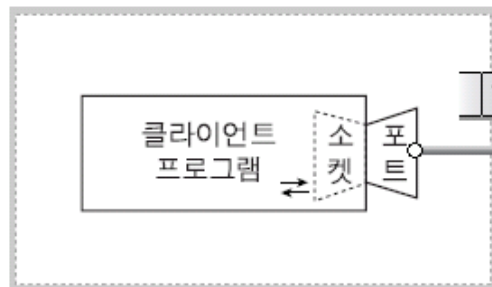
컴퓨터 A



컴퓨터 B

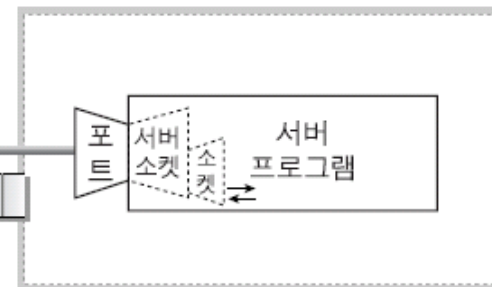


컴퓨터 A



- 4) 클라이언트 프로그램은 연결 요청에 사용한 소켓을 통해 데이터를 주고 받습니다

컴퓨터 B



- 3) 서버 프로그램은 서버 소켓이 만든 소켓을 통해 데이터를 주고받습니다

# TCP/IP 통신 프로그램의 작성 방법

---

## ❖ 소켓을 이용한 통신 : 서버 프로그램

- 서버 소켓을 생성하고 사용하는 방법

1) ServerSocket 객체를 생성합니다.

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



포트 번호

# TCP/IP 통신 프로그램의 작성 방법

---

## ❖ 소켓을 이용한 통신 : 서버 프로그램

- 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

- 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

### ○ 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

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

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

↑  
서버 프로그램이 있는  
컴퓨터의 IP 주소

↑  
서버 프로그램이  
열어 놓은 포트 번호

# TCP/IP 통신 프로그램의 작성 방법

---

## ❖ 소켓을 이용한 통신 : 클라이언트/서버 프로그램

- 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