

4. IP주소 관련 클래스

서버/클라이언트

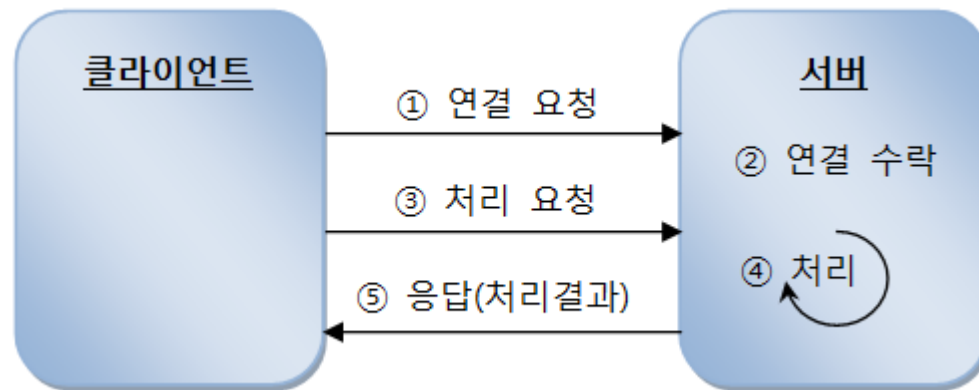
IP주소와 포트

IP주소 관련 클래스



1. 서버와 클라이언트

- ❖ 서버: 서비스를 제공하는 프로그램
 - 웹 서버, FTP서버, DBMS, 메신저 서버
 - 클라이언트의 연결을 수락하고, 요청 내용 처리한 후 응답 보내는 역할
- ❖ 클라이언트: 서비스를 받는 프로그램
 - 웹 브라우저, FTP 클라이언트, 메신저
 - 네트워크 데이터를 필요로 하는 모든 애플리케이션이 해당(모바일 앱 포함)

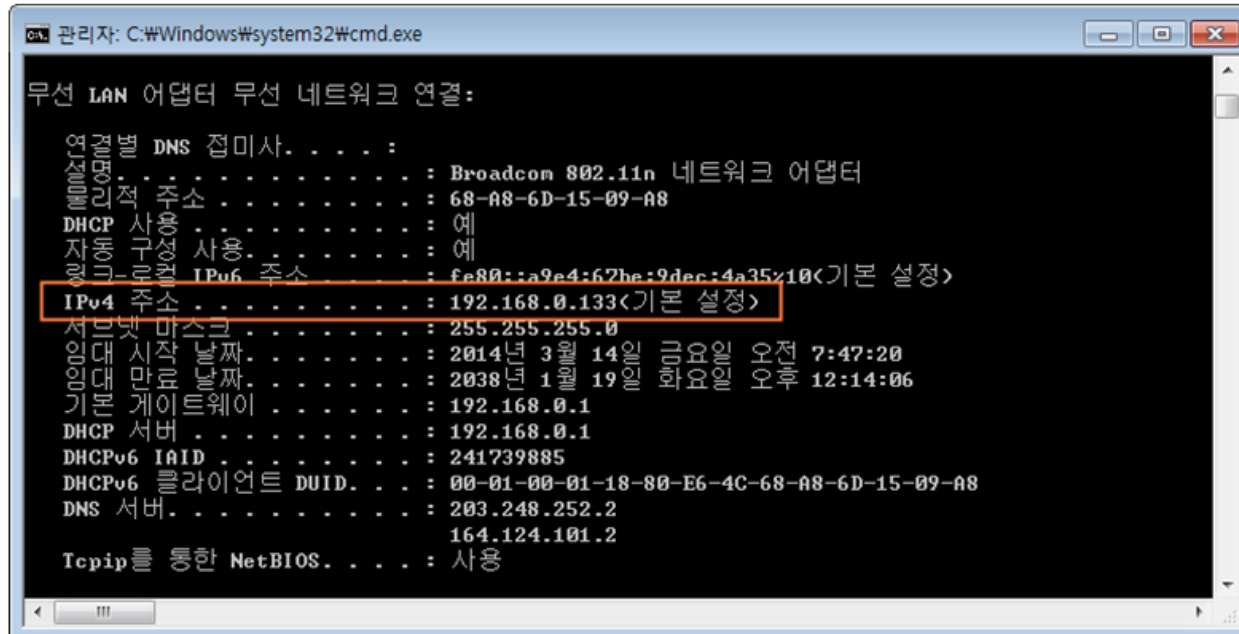


2. IP 주소와 포트

❖ IP(Internet Protocol) 주소

- 네트워크상에서 컴퓨터를 식별하는 번호
- 네트워크 어댑터(랜 (Lan) 카드) 마다 할당
- IP 주소 확인 법 – 명령 프롬프트 (cmd.exe) 사용
- XXX.XXX.XXX.XXX 형식으로 표현 (xxx는 0~255 사이의 정수)

```
C:\W>ipconfig /all
```



```
관리자: C:\Windows\system32\cmd.exe

무선 LAN 어댑터 무선 네트워크 연결:

연결별 DNS 접미사. . . . . :
설명. . . . . : Broadcom 802.11n 네트워크 어댑터
물리적 주소. . . . . : 68-A8-6D-15-09-A8
DHCP 사용. . . . . : 예
자동 구성 사용. . . . . : 예
링크-로컬 IPv6 주소. . . . . : fe80::a2e4:67be:9dec:4a35%10<기본 설정>
IPv4 주소. . . . . : 192.168.0.133<기본 설정>
서브넷 마스크. . . . . : 255.255.255.0
임대 시작 날짜. . . . . : 2014년 3월 14일 금요일 오전 7:47:20
임대 만료 날짜. . . . . : 2038년 1월 19일 화요일 오후 12:14:06
기본 게이트웨이. . . . . : 192.168.0.1
DHCP 서버. . . . . : 192.168.0.1
DHCPv6 IAID. . . . . : 241739885
DHCPv6 클라이언트 DUID. . . : 00-01-00-01-18-80-E6-4C-68-A8-6D-15-09-A8
DNS 서버. . . . . : 203.248.252.2
                  164.124.101.2
Tcpip를 통한 NetBIOS. . . . : 사용
```

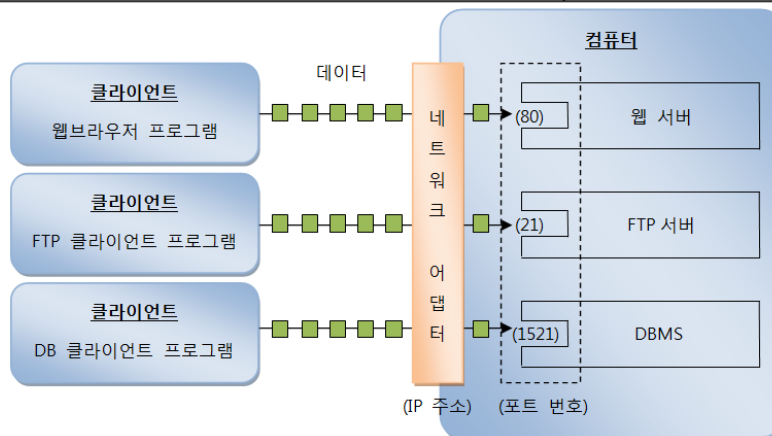


2. IP 주소와 포트

■ 포트(Port)

- 같은 컴퓨터 내에서 프로그램을 식별하는 번호
- 클라이언트는 서버 연결 요청 시 IP 주소와 Port 같이 제공
- 0~65535 범위의 값을 가짐
- 포트 범위는 세 가지로 구분

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



3. IP주소 관련 클래스-InetAddress

❖ java.net.InetAddress

- public final class InetAddress extends Object implements Serializable
 - IP 주소를 캡슐화하여 표현한 클래스
 - 로컬 컴퓨터의 IP 주소
 - 도메인 이름을 DNS에서 검색한 후 IP 주소를 가져오는 기능 제공
- 필드
 - `hostName` → www.oreilly.com 과 같이 호스트 이름을 저장
 - `address` → 32 비트 IP 주소를 저장
- InetAddress 객체 만들기
 - 초기화된 InetAddress 객체를 얻기 위한 3개의 정적 메소드
 - `getByName()`
 - `getAllByName()`
 - `getLocalHost()`



3. IP주소 관련 클래스

❖ `getByName()`

- `public static InetAddress InetAddress.getByName(String hostName)` Throws `UnknownHostException`
- 호스트 이름을 인자로 DNS를 사용하여 해당 호스트 IP주소를 얻음
- 호스트를 찾을 수 없다면 `UnknownHostException` 발생

■ 사용방법

- `InetAddress address = InetAddress.getByName("www.oreilly.com");`
- `InetAddress address = InetAddress.getByName("204.148.40.9");`
 - IP 주소를 인자로 넘길 수 있음



InetAddress

❖ 사용 예:

```
import java.net.*;
public class OReillyByName {
    public static void main (String[] args) {
        try {
            InetAddress address =
                InetAddress.getByName("www.oreilly.com");
            System.out.println(address);
        }
        catch (UnknownHostException e) {
            System.out.println("Could not find www.oreilly.com");
        }
    }
}
```

```
% java OReillyByName
www.oreilly.com/204.148.40.9
```



InetAddress

❖ getAllByName()

- `public static InetAddress[] InetAddress.getAllByName (String hostName)` throws `UnknownHostException`
 - 호스트의 이름을 인자로 대응하는 모든 주소를 포함하는 배열을 리턴
- 사용 방법
 - `InetAddress[] addresses =
InetAddress.getAllByName("www.apple.com");`
 - 하나의 호스트 이름에 여러 개의 IP 가 있을 경우 모두 찾음
 - 호스트를 찾을 수 없을 경우 `UnknownHostException` 발생



InetAddress

❖ 사용예:

```
try {  
    InetAddress[] addresses =  
        InetAddress.getAllByName("www.microsoft.com");  
    for (int i = 0; i < addresses.length; i++) {  
        System.out.println(addresses[i]);  
    }  
} catch (UnknownHostException e) {  
    System.out.println("Could not find www.microsoft.com");  
}
```

```
% java AllAddressesOfMicrosoft  
www.microsoft.com/207.46.131.15  
www.microsoft.com/207.46.131.137  
www.microsoft.com/207.46.130.14  
www.microsoft.com/207.46.130.149  
www.microsoft.com/207.46.130.150  
www.microsoft.com/207.46.131.13
```

InetAddress

❖ getLocalHost()

- `public static InetAddress InetAddress.getLocalHost()` Throws `UnknownHostException`
- 현재 메소드가 시행되고 있는 호스트의 주소를 `address` 필드 값으로 가지는 `InetAddress` 클래스를 리턴
- 로컬 호스트의 주소를 찾을 수 없을 경우 `UnknownHostException` 을 발생

■ 사용 방법

```
InetAddress thisComputer = InetAddress.getLocalHost( );
```



InetAddress

❖ 사용 예:

```
try {  
    InetAddress address = InetAddress.getLocalHost( );  
    System.out.println(address);  
}  
catch (UnknownHostException e) {  
    System.out.println("Could not find this computer's address.");  
}
```

```
% java MyAddress  
titan.oit.unc.edu/152.2.22.14
```



❖ Getter 메소드

- `public String getHostName()`
- IP 주소로 해당하는 호스트의 이름 찾기
- 호스트의 이름을 찾지 못할 경우 IP 주소를 리턴
- 사용 방법
 - `InetAddress machine = InetAddress.getLocalHost();`
 - `String localhost = machine.getHostName();`
 - 로컬 DNS의 동작 방법에 따라 아래와 같이 나타날 수 있다.
 - `titan.oit.unc.edu` → `titan`



- public String GetHostAddress()
 - InetAddress 객체에서 IP 주소를 리턴

```
try {  
    InetAddress me = InetAddress.getLocalHost( );  
    String dottedQuad = me.getHostAddress( );  
    System.out.println("My address is " + dottedQuad);  
}  
catch (UnknownHostException e) {  
    System.out.println("I'm sorry. I don't know my own address.");  
}
```

```
% java MyDottedQuadAddress  
My address is 152.2.22.14.
```



❖ public byte[] getAddress()

■ 네트워크 바이트 순서대로 IP 주소의 바이트 배열을 리턴

■ 사용방법

- InetAddress me = InetAddress.getLocalHost();
- byte[] address = me.getAddress();

■ 바이트를 정수형으로 변환

- int unsignedByte = signedByte < 0 ? signedByte + 256 : signedByte;



❖ 객체 메소드

■ `public boolean equals(Object o)`

- 동일한 객체인지 검증 하는 메소드
- 두 객체가 모두 `InetAddress` 의 인스턴스이고 같은 IP 주소를 가진다면
- → 동일한 `InetAddress` 객체라 할 수 있음

■ `public int hashCode()`

- `InetAddress` 객체가 해쉬 테이블에서 키로 사용될 때 필요한 정수 값을 리턴
- `Java.util.Hashtable`의 메소드에 의해 호출
- `Address` 필드의 값 반환
- 16바이트 주소로의 수정이 어려움



❖ 객체 메소드

■ Public String toString()

- 호스트의 이름이나 IP 주소에 대한 문자열을 리턴
- 만들어지는 문자열
 - 호스트 이름 / 점으로 구분된 4자리 Ip 주소

■ 사용방법

- `InetAddress thisComputer = InetAddress.getLocalHost();`
- `String address = thisComputer.toString();`

