

알기 쉬운 J A V A

9th edition

20장 네트워크

Section 1 네트워킹의 개요와 java.net 패키지

Section 2 인터넷 주소와 URL

Section 3 TCP 소켓

Section 4 UDP 소켓



• 학습 목표

9th edition

- 네트워킹의 개념에 대해 학습합니다.
- 인터넷의 주소와 URL을 네트워크를 통해 사용하는 방법을 학습합니다.
- 연결성 통신 방법인 TCP 소켓에 대해 학습합니다. 두 개의 프로그램이 연결성 통신 방법을 사용하여 통신하는 프로그램을 작성합니다.
- 비연결성 통신 방법인 UDP 소켓에 대해 학습합니다. 두 개의 프로그램이 비연결성 통신 방법을 사용하여 통신하는 프로그램을 작성합니다.



- **프로토콜**

- 컴퓨터 상호 간에 통신을 위한 규약으로 정의

- **TCP/IP 프로토콜**

- 4개의 기능 계층들로 구성

1-1 TCP/IP(Transmission Control Protocol/Internet Protocol)



그림 17-1 TCP/IP 계층 구조



1-2 TCP와 UDP

● TCP/IP의 전달 계층은 크게 두 가지로 구분

- 연결형인 TCP(Transmission Control Protocol)
- 비연결형인 UDP(User Datagram Protocol)

● TCP

- 연결성 통신 방식으로 전화처럼 먼저 수신자와 연결을 설정한 다음 정보를 주고받는 방식을 의미

● UDP

- 비연결성 통신 방식으로 편지처럼 보내는 사람에 의해 보내지면 받는 사람은 그 편지를 받아야 편지가 도착한 사실을 알게 되는 경우



1-3 소켓(socket)

● 소켓(socket)

- 전기선의 소켓과 유사
- 컴퓨터가 연결된 통신의 끝점을 의미
- 상호 연결된 응용 프로그램들 사이의 안정된 정보 교환을 제공하고 있는 매체
- TCP/IP소켓 : TCP(Transmission Control Protocol) 소켓, UDP(User Datagram Protocol) 소켓



1-4 포트(port)

● 포트

- 통신선을 통해 수신되는 데이터가 컴퓨터 내의 여러 통신 프로그램 중에서 하나의 프로그램에 전달되도록 하기 위한 번호
- 인터넷을 통하여 전달되는 정보들은 목적지 컴퓨터의 주소(32비트)와 16비트의 포트번호로 구성

- 예 : 203.233.51.1:8088

1-5 java.net 패키지

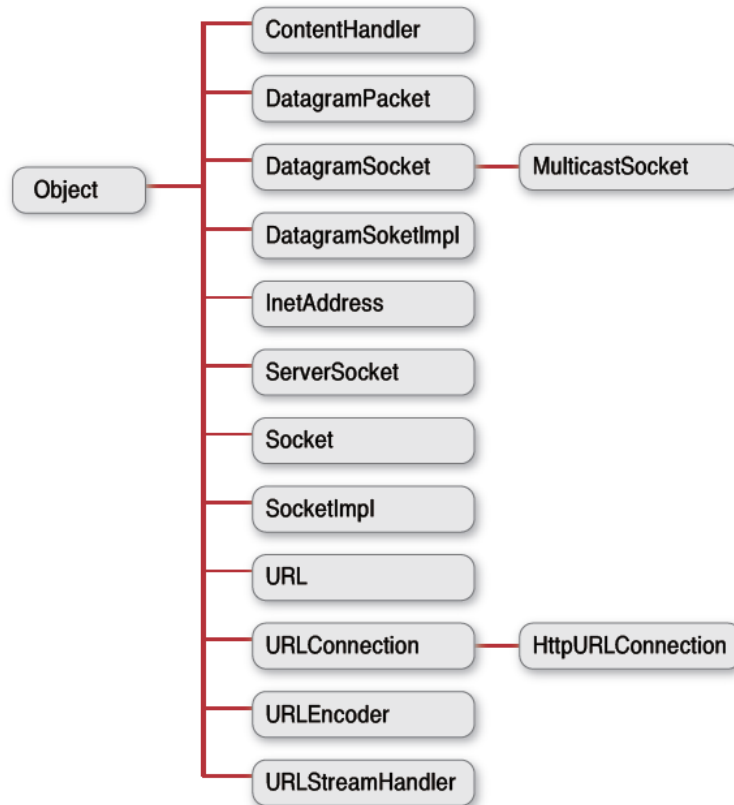


그림 17-2 java.net 패키지의 클래스

● 인터넷의 주소

- IP(Internet Protocol) 주소 : 숫자로 구성된 주소
 - (예 : 134.23.33.200)
- 도메인(Domain) 이름 : 문자중심
 - (예 : www.ehan.co.kr)

● 편리함으로 인하여 대부분의 사용자는 도메인 이름을 사용

URL Uniform Resource Locator은 웹 World Wide Web에서 사용하는 주소입니다.

【 형식 】 URL의 형식

```
protocol://host:port/filename(경로포함)
```

2-1 InetAddress 클래스

● InetAddress 클래스

- 인터넷 주소에 관한 정보를 가짐
- 주소와 관련되 다양한 메소드를 제공

표 17-1 InetAddress 클래스의 주요 메소드

메소드	설명
<code>boolean equals(InetAddress other)</code>	현 객체가 other 객체와 같은 주소를 가지면 true, 아니면 false를 반환
<code>byte[] getAddress()</code>	주소를 나타내는 4개의 요소를 가진 바이트 배열을 반환
<code>String getHostAddress()</code>	주소 정보를 나타내는 문자열을 반환
<code>String getHostName()</code>	컴퓨터 이름을 나타내는 문자열을 반환
<code>static InetAddress getLocalHost() throws UnknownHostException</code>	현재 컴퓨터를 나타내는 InetAddress 객체를 반환
<code>static InetAddress getByName (String hostName) throws UnknownHostException</code>	hostName으로 지정된 컴퓨터를 나타내는 InetAddress 객체를 반환
<code>static InetAddress[] getAllByName (String hostName) throws UnknownHostException</code>	hostName으로 지정된 모든 컴퓨터(하나의 도메인 이름으로 여러 대의 컴퓨터를 사용하는 경우)를 나타내는 InetAddress 객체들의 배열을 반환

2-1 InetAddress 클래스

예제 17.1

AddressTest.java

```

01: import java.net.*;
02: public class AddressTest {
03:     public static void main(String args[]) throws UnknownHostException {
04:         InetAddress Address = InetAddress.getLocalHost(); ← 현재 컴퓨터의
05:         System.out.println ←                               InetAddress 객체 생성
06:         ("로컬 컴퓨터의 이름 : " + Address.getHostName()); ← 컴퓨터의 이름 추출
07:         System.out.println ←                               컴퓨터의 IP 주소 추출
08:         ("로컬 컴퓨터의 IP 주소 : " + Address.getHostAddress()); ←
09:         Address = InetAddress.getByName("java.sun.com"); ←
10:         System.out.println ←                               도메인 이름으로 InetAddress 객체 생성
11:         ("java.sun.com 컴퓨터의 이름과 IP 주소 : " + Address);
12:         InetAddress all[] = ←                               www.daum.net 컴퓨터의 주소를 배열로 생성
13:         InetAddress.getAllByName("www.daum.net"); ←
14:         for (int i=0; i < all.length; i++)
15:             System.out.println(all[i]);
16:     }
17: }

```

실행 결과

```

로컬 컴퓨터의 이름 : cskim-PC
로컬 컴퓨터의 IP 주소 : 177.17.6.89
java.sun.com 컴퓨터의 이름과 IP 주소 : java.sun.com/192.9.162.55
www.daum.net/211.115.77.212
www.daum.net/211.115.77.213
www.daum.net/211.32.117.30
www.daum.net/211.115.77.211
www.daum.net/222.231.51.77
www.daum.net/222.231.51.78
www.daum.net/211.115.115.212
www.daum.net/222.231.51.40
www.daum.net/211.115.77.214
www.daum.net/211.115.115.211

```

2-2 URL 클래스

● URL 클래스

- Web에서 사용하는 URL에 관한 정보를 가짐

[형식] URL 클래스의 생성자

`URL(String protocol, String host, int port, String file)` throws `MalformedURLException`

`URL(String protocol, String host, String file)` throws `MalformedURLException`

`URL(String urlString)` throws `MalformedURLException`

`protocol, host, port, file` : URL의 구성요소

`urlString` : 모든 요소를 표현한 문자열

2-2 URL 클래스

표 17-2 URL 클래스의 주요 메소드

메소드	설명
String getFile()	URL의 파일 이름을 반환
String getHost()	URL의 호스트 이름을 반환
String getPort()	URL의 포트번호를 반환, 묵시적인(default) 포트일 경우 -1을 반환
String getProtocol()	URL의 프로토콜 이름을 반환
String toExternalForm()	전체 URL의 문자열 객체를 반환
URLConnection.openConnection() throws IOException	지정된 URL과 연결 후 URLConnection 객체를 반환
InputStream openStream() throws IOException	지정된 URL로부터 정보를 읽어들이기 위한 객체를 반환

예제 17.2

URLTest.java

```

01: import java.net.*;
02: public class URLTest {
03:     public static void main(String args[]) throws
04:         MalformedURLException {
05:         URL kbs = new URL
06:             ("http://www.kbs.co.kr/aboutkbs/history.html");
07:         System.out.println("프로토콜: " + kbs.getProtocol());
08:         System.out.println("포트: " + kbs.getPort());
09:         System.out.println("호스트: " + kbs.getHost());
10:         System.out.println("파일(경로포함): " + kbs.getFile());
11:         System.out.println("전체 URL: " + kbs.toExternalForm());
12:     }
13: }

```

특정 URL을 지정하여 객체 생성

프로토콜을 출력

포트를 출력

실행 결과

프로토콜: http

포트: -1

호스트: www.kbs.co.kr

파일(경로포함): /aboutkbs/history.html

전체 URL: http://www.kbs.co.kr/aboutkbs/history.html

2-3 URLConnection 클래스

● URLConnection 클래스

- 원격지 자원의 속성(attribute)을 알아내기 위한 목적으로 많이 사용
- 상대 컴퓨터와 연결된 상태에서 원격지 자원의 속성을 파악
- 원격지 파일을 읽어오는 다양한 메소드를 제공

표 17-3 URLConnection 클래스의 주요 메소드

메소드	설명
int getLength()	해당 문서의 길이를 바이트 수로 반환
String getContentType()	해당 문서의 타입을 반환
long getDate()	해당 문서의 생성 날짜를 반환
long getExpiration()	해당 문서의 파기 날짜를 반환
long getLastModified()	해당 문서의 마지막 수정 날짜를 반환
InputStream getInputStream() throws IOException	원격지로부터 정보를 읽어들이기 위한 InputStream 객체를 생성하여 반환

2-3 URLConnection 클래스

예제 17.3

URLConnectionTest.java

```

01: import java.net.*;
02: import java.io.*;
03: public class URLConnectionTest {
04:     public static void main(String args[]) throws Exception {
05:         URL kbs = new URL
06:             ("http://www.kbs.co.kr/aboutkbs/history.html");
07:         URLConnection kbsCon = kbs.openConnection(); ←----- URL 객체를 이용하여
08:         System.out.println                                URLConnection 객체 생성
09:             ("문서의 타입: " + kbsCon.getContentType());
10:         System.out.println("== 문서의 내용 ==");
11:         InputStream input = kbsCon.getInputStream(); ←----- 입력을 위한 객체 생성
12:         int i = 500;
13:         int c;
14:         while (((c = input.read()) != -1) && (--i > 0)) { ←----- 입력된 내용을 출력
15:             System.out.print((char) c);
16:         }
17:         input.close();
18:     }
19: }

```


2-3 URLConnection 클래스

실행 결과

```

문서의 타입: text/html
== 문서의 내용 ==
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<title>KBS i ' ??? ?</title>
<link href="http://img.kbs.co.kr/cms/aboutkbs/image/style.css" rel="stylesheet" type="text/
css">
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_reloadPage(init) { //reloads the window if Nav4 resized
  if (init==true) with (navigator) {if ((appName=="Netscape")&&(parseInt(appVersion)==4)) {
    document.MM_pgW=innerWidth; document.MM_pgH=innerHeight; onresi
  
```

● ServerSocket 클래스

- 서버 측에서 실행되는 응용 프로그램 작성을 위해 사용

【 형식 】 ServerSocket 클래스의 생성자

`ServerSocket(int port) throws IOException`

port : 요청을 받아들일 포트 번호

표 17-4 ServerSocket 클래스의 주요 메소드

메소드	설명
<code>Socket accept() throws IOException</code>	클라이언트의 요청을 받아들인 다음, Socket 클래스 객체를 반환합니다.
<code>void close() throws IOException</code>	서버 소켓을 닫습니다.

● Socket 클래스

- 클라이언트와 서버 사이에 실질적인 정보 교환을 위해 사용

【 형식 】 Socket 클래스의 생성자

```
Socket(String hostName, int port) throws UnknownHostException, IOException
```

hostName, port : 연결을 요청할 컴퓨터의 주소와 포트 번호

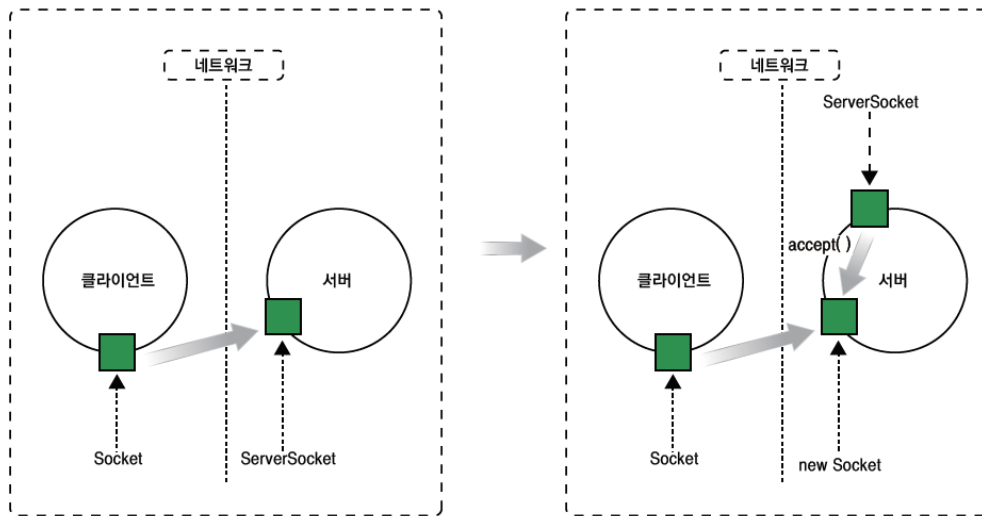


그림 17-3 TCP 소켓의 작동

표 17-5 Socket 클래스의 주요 메소드

메소드	설명
OutputStream getOutputStream() throws IOException	현재의 소켓과 관련된 OutputStream 객체를 반환
void close() throws IOException	소켓을 닫습니다.
InetAddress getInetAddress()	현재 소켓에 연결된 컴퓨터의 주소를 반환
InetAddress getLocalAddress()	현재 소켓을 사용하고 있는 컴퓨터의 주소를 반환
int getPort()	현재 소켓에 연결된 컴퓨터의 포트번호를 반환
int getLocalPort()	현재 소켓이 사용하고 있는 포트번호를 반환
InputStream getInputStream() throws IOException	현재의 소켓과 관련된 InputStream 객체를 반환

● ServerSocket과 Socket 클래스를 이용한 클라이언트-서버 간의 통신 순서

- 서버 측

- ① 서버 소켓 객체를 ServerSocket 클래스로부터 생성한 다음 클라이언트의 요청을 기다린다(accept() 메소드).
- ② 클라이언트의 요청이 발생되면 요청을 받아들여 Socket 객체를 생성한다.
- ③ ocket 객체를 이용하여 입출력을 위한 스트림을 생성한다.
- ④ 통신을 수행한다.
- ⑤ 소켓을 닫는다.

- 클라이언트 측

- ① 연결을 원하는 서버의 주소와 포트번호로 Socket 객체를 생성한다.
- ② Socket 객체를 이용하여 입출력을 위한 스트림을 생성한다.
- ③ 통신을 수행한다.
- ④ 소켓을 닫는다.

예제 17.4

ServerSide.java

```

01: import java.io.*;
02: import java.net.*;
03: class ServerSide {
04:     public static void main(String args[]) throws Exception {
05:         int port = Integer.parseInt(args[0]);
06:         int times = Integer.parseInt(args[1]);
07:         ServerSocket ss = new ServerSocket(port);
08:         int i = 1;
09:         while( i <= times) {
10:             Socket s = ss.accept();
11:             OutputStream os = s.getOutputStream();
12:             DataOutputStream dos = new DataOutputStream(os);
13:             for(int j = 1 ; j <= 10 ; j++)
14:                 dos.writeInt(j);
15:             s.close();
16:             ++i;
17:         }
18:         ss.close();
19:     }
20: }

```

실행 결과

이 프로그램을 실행하면 서버 프로그램은 대기 상태가 됩니다. 지정된 횟수만큼의 클라이언트 요청을 처리하면 프로그램이 종료됩니다.

← 포트번호로 ServerSocket 객체 생성

← 지정된 횟수만큼 클라이언트 요청 처리

← 클라이언트 요청을 기다려 Socket 객체를 생성

← 기본 자료형 데이터 출력을 위한 객체 생성

← 정수를 출력

← 바이트 스트림 출력을 위한 객체를 Socket 객체로 생성

예제 17.5

ClientSide.java

```

01: import java.io.*;
02: import java.net.*;
03: public class ClientSide {
04:     public static void main(String args[]) throws Exception {
05:         String server = args[0];
06:         int port = Integer.parseInt(args[1]);
07:         Socket c = new Socket(server, port);
08:         InputStream is = c.getInputStream();
09:         DataInputStream dis = new DataInputStream(is);
10:         for(int i=1 ; i <= 10 ; i++) {
11:             int j = dis.readInt();
12:             System.out.println
13:                 ("서버로부터 받은 데이터 " + j + " 출력");
14:         }
15:         c.close();
16:     }
17: }
  
```

서버의 주소와 포트번호로
Socket 객체 생성

정수를 입력받기 위한
스트림 객체 생성

정수를 읽는다

실행 결과

상대방 컴퓨터의 주소와 포트번호를 지정하여 프로그램을 실행하여 다음과 같은 결과를 출력하였습니다. 한 대의 컴퓨터에서 서버와 클라이언트 프로그램을 모두 실행시키는 경우, 컴퓨터의 주소로 127.0.0.1(자신의 컴퓨터를 의미) 또는 localhost로 지정하면 됩니다. 포트번호는 4자리수의 포트번호를 자유롭게 사용하면 됩니다(예 : 1111).

서버로부터 받은 데이터 1 출력
 서버로부터 받은 데이터 2 출력
 서버로부터 받은 데이터 3 출력
 서버로부터 받은 데이터 4 출력
 서버로부터 받은 데이터 5 출력
 서버로부터 받은 데이터 6 출력
 서버로부터 받은 데이터 7 출력
 서버로부터 받은 데이터 8 출력
 서버로부터 받은 데이터 9 출력
 서버로부터 받은 데이터 10 출력

● UDP(User Datagram Protocol)

- TCP 소켓에 비해 신뢰성과 안정성 측면에서 떨어짐
- 연결을 설정하지 않으므로 네트워크에 부담을 주지 않는다는 장점

● DatagramPacket 클래스

- 응용 프로그램들이 주고받을 데이터와 관련된 클래스
- 실제 데이터의 전송은 DatagramSocket 클래스에 의해 이뤄짐

【 형식 】 DatagramPacket 클래스의 생성자

`DatagramPacket(byte[] buffer, int size)`

`DatagramPacket(byte[] buffer, int size, InetAddress ia, int port)`

buffer : 송수신될 데이터가 저장되어 있는 배열

size : 배열의 크기

ia : 상대방 컴퓨터의 주소(InetAddress 객체)

port : 상대방 컴퓨터의 포트 번호

표 17-6 DatagramPacket 클래스의 주요 메소드

메소드	설명
<code>InetAddress getAddress()</code>	수신 응용 프로그램에서 사용합니다. 정보를 보낸 컴퓨터의 주소를 반환
<code>byte[] getData()</code>	패킷으로부터 데이터를 읽어들이어 바이트 배열로 반환
<code>int getLength()</code>	패킷의 바이트 수를 반환
<code>int getPort()</code>	포트번호를 반환
<code>void setAddress(InetAddress ia)</code>	ia를 주소로 설정
<code>void setData(byte buffer[])</code>	buffer의 내용을 패킷의 데이터로 설정
<code>void setLength(int size)</code>	패킷의 크기를 size로 설정
<code>void setPort(int pt)</code>	포트번호를 pt 값으로 설정

● DatagramSocket 클래스

- 실제 정보를 주고받기 위한 기능을 제공하는 클래스
- 두 개의 생성자를 제공

【 형식 】 DatagramSocket 클래스의 생성자

`DatagramSocket()` throws `SocketException`

`DatagramSocket(int port)` throws `SocketException`

port : 소켓이 사용할 포트번호.

표 17-7 DatagramSocket 클래스의 주요 메소드

메소드	설명
<code>void receive(DatagramPacket dgram)</code> throws <code>IOException</code>	현재의 소켓으로부터 정보를 읽어 dgram 패킷에 저장
<code>void send(DatagramPacket dgram)</code> throws <code>IOException</code>	현재의 소켓을 통하여 dgram 패킷을 전송
<code>void close()</code> throws <code>IOException</code>	소켓을 닫습니다.

예제 17.6

UDPReceiver.java

```

01: import java.net.*;
02: public class UDPReceiver {
03:     public static void main(String args[]) throws Exception {
04:         int port = Integer.parseInt(args[0]);
05:         int times = Integer.parseInt(args[1]);
06:         DatagramSocket ds = new DatagramSocket(port); ← 특정 포트를 지정하여
07:         int i=1 ;                                         객체 생성
08:         while( i <= times) {
09:             byte buffer[] = new byte[30]; ← 바이트 배열(30바이트 크기) 생성
10:             DatagramPacket dp = ←
11:                 new DatagramPacket(buffer, buffer.length); ← 패킷을 생성(30바이트 크기)
12:             ds.receive(dp); ← 대기하다가 편지가 오면 패킷에 저장
13:             String str = new String(dp.getData()); ← 패킷에 저장된 데이터를 추출
14:             System.out.println("수신된 데이터 : " + str);
15:             ++i;
16:         }
17:     }
18: }

```

실행 결과

UDPReceiver 프로그램에서 보낸 내용이 출력되었습니다.

수신된 데이터 : 자바는 쉽다
 수신된 데이터 : 아니다 어렵다
 수신된 데이터 : 자바는 정말 쉽다
 수신된 데이터 : 열심히 하면 쉽다

예제 17.7

UDPSender.java

```

01: import java.net.*;
02: public class UDPSender {
03:     public static void main(String args[]) throws Exception {
04:         DatagramSocket ds = new DatagramSocket();
05:         InetAddress ia = InetAddress.getByName(args[0]);
06:         int port = Integer.parseInt(args[1]);
07:         byte buffer[] = args[2].getBytes();
08:         DatagramPacket dp =
09:             new DatagramPacket
10:                 (buffer, buffer.length, ia, port);
11:         ds.send(dp);
12:     }
13: }
  
```

DatagramSocket 객체를 생성(주소 지정 안 함)
 InetAddress 객체 생성 (주소를 가진 객체)
 세 번째 매개변수의 값(편지내용)을 바이트 배열로 변환
 패킷 객체 생성 (주소, 포트, 내용)
 패킷을 보낸다

실행 결과

위 프로그램을 한 대의 컴퓨터에서 실행시키기 위해 다음과 같이 3개의 매개변수를 지정하였습니다. 4 번을 실행하였습니다.

```

127.0.0.1 1111 자바는 쉽다
127.0.0.1 1111 아니다어렵다
127.0.0.1 1111 자바는 정말 쉽다
127.0.0.1 1111 열심히하면 쉽다
  
```

■ 학습정리

● 네트워킹의 개요와 java.net 패키지

- ① 자바는 네트워킹 관련 클래스들을 java.net 패키지로 제공하고 있다.
- ② 인터넷에서는 TCP/IP 프로토콜을 표준 프로토콜로 사용하고 있다.
- ③ 통신 방법에는 연결성 통신 방법인 TCP 방법과 비연결성 통신 방법인 UDP 방법이 있다. 소켓은 컴퓨터가 네트워크에 연결된 끝점을 의미한다.
- ④ 포트는 하나의 컴퓨터에 여러 개 존재하는 논리적인 개념이다. 서로 다른 포트를 이용하여 컴퓨터에 서로 다른 일을 시킬 수 있다.

● 인터넷 주소와 URL

- ① 인터넷의 주소에는 컴퓨터 상호 간에 사용하는 숫자로 구성된 IP(Internet Protocol) 주소가 있고(예 : 134.23.33.200), 우리가 많이 사용하는 문자 중심의 도메인(Domain) 이름(예 : www.hollywood.com)이 있다.
- ② InetAddress 클래스는 인터넷의 주소 정보를 가진 클래스이다.
- ③ URL 클래스와 URLConnection 클래스를 이용하여 인터넷으로부터 정보를 얻어낼 수 있다.

■ 학습정리

● TCP 소켓

- ① 자바는 클라이언트-서버 응용 프로그램 작성을 위한 ServerSocket 클래스와 Socket 클래스를 제공하고 있다.
- ② TCP 소켓은 상호 연결된 상태에서 신뢰성 있는 통신을 제공한다.
- ③ ServerSocket 클래스는 서버 측에서 사용되며 실제 통신을 수행하지 않는다. 실제 통신은 Socket 객체에 의해 이루어진다.

● UDP 소켓

- ① UDP(User Datagram Protocol)는 TCP 소켓에 비해 신뢰성과 안정성 측면에서 뒤지지만, 연결을 설정하지 않으므로 네트워크에 부담을 주지 않는다는 장점이 있다.
- ② 자바는 UDP를 지원하기 위해 DatagramPacket, DatagramSocket 클래스를 제공하고 있다.
- ③ DatagramPacket 클래스는 응용 프로그램들이 주고받을 데이터와 관련된 클래스이고, 실제 데이터의 전송은 DatagramSocket 클래스에 의해 이루어진다.

알기 쉽게 해설한 J A V A

9th edition

**수고하셨습니다.
제출 연습 문제를 잘 풀어서
제출해주시기 바랍니다**

