

# (디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용SW개발자 양성과정

---

2021년 6월 9일  
[ 23일차 복습 ]

- 수강생 : 김 민 규
- 강의장 : 강 남 C
- 수강 기간 : 2021. 05. 07 ~ 2021. 12. 08
- 수강 시간 : 15:30 ~ 22:00
- 이상훈 강사님 | 이은정 취업담임



## ▶ 내용 :

```
public class SortingTest {
    public static void main(String[] args) {
        String[] sample = {"I", "walk", "the", "line", "Apple", "hit", "me", "Ground", "attack", "you"};

        List<String> list = Arrays.asList(sample);

        // 정렬 법칙(대문자 우선, 그 다음 소문자)
        Collections.sort(list);

        System.out.println(list);

        Integer[] numbers = {1, 2, 3, 100, 77, 2342, 2342354, 345, 12323, 12, 4};

        List<Integer> numList = Arrays.asList(numbers);

        Collections.sort(numList);

        System.out.println(numList);

        Set fruits = new HashSet();

        fruits.add("strawberry");
        fruits.add("watermelon");
        fruits.add("grape");
        fruits.add("orange");
        fruits.add("apple");
        fruits.add("banana");

        List fruitsList = new ArrayList(fruits);

        Collections.sort(fruitsList);

        System.out.println(fruitsList);
    }
}
```

## 주요 내용

--

Array.Sort  
Collection.sort

오름차순 정렬

--

▶ 내용 : 스레드 \_ 반드시 알아야할 개념

주요 내용

/\* 여기서 가져가야할 주요 개념

1. Thread를 활용하는 이유는 성능을 빠르게 만들기 위함이다
  2. 비동기 패턴(Asynchronous Pattern)이란 전부 Thread를 기반으로 한다.
  3. 자바 스크립트 또한 Multi Thread 모델을 지원한다(자체적으로)  
(이건 최신 자바스크립트 ECMA 6 부터 서포트인것 같음) - Promise를 활용하여 증명
  4. Thread를 사용할 때는 Critical Section에 대한 방어가 무엇보다도 중요하다(데이터 무결성)
  5. 또한 스레드는 비동기 처리를 하기 때문에 데이터의 완전한 전송을 보장하지 못할 수도 있다.  
(말이 좀 어려운데 이 부분은 자바스크립트의 Promise를 통해 살펴볼 예정)
- ex) 전화 통화: 동기 처리  
왜 ? 친구한테 전화를 걸었음. 친구가 통화 허용을 안하면 통화가 안됨
- ex) 카카오톡 메시지: 비동기 처리  
왜 ? 상대방이 확인하던 안하던 난 보낸다.  
나는 니가 뭘 하던 내 할 일을 하겠다.

--  
--

## ▶ 내용 : 네트워크

```
+ import java.net.MalformedURLException;
+ import java.net.URL;
+
+ public class NetworkUrlTest {
+     // Malform 이라는것이 악성 코드에 해당해서
+     // 이상한 URL로 링크를 태워서 공격을 할 수 있기 때문에 그것에 대한 방어 조치라 보면 됨
+     // www.daum.net, http://www.daum.net
+     // URL을 반드시 후자로 줘야 합니다.
+     // 이유는 www.daum.net 으로 하면 위와 같이 악성코드 공격이 가능함
+     public static void main(String[] args) throws MalformedURLException {
+         URL myURL = new URL("http://www.loanconsultant.or.kr/source/index.jsp?t=20191216");
+
+         // Protocol: HTTP(웹 애플리케이션 전용 프로토콜입니다)
+         System.out.println("Protocol = " + myURL.getProtocol());
+         System.out.println("authority = " + myURL.getAuthority());
+         System.out.println("host = " + myURL.getHost());
+         System.out.println("port = " + myURL.getPort());
+         System.out.println("path = " + myURL.getPath());
+         System.out.println("query = " + myURL.getQuery());
+         System.out.println("filename = " + myURL.getFile());
+         System.out.println("ref = " + myURL.getRef());
+     }
+ }
```

## 주요 내용

--

악성코드 방어조치

- Malform

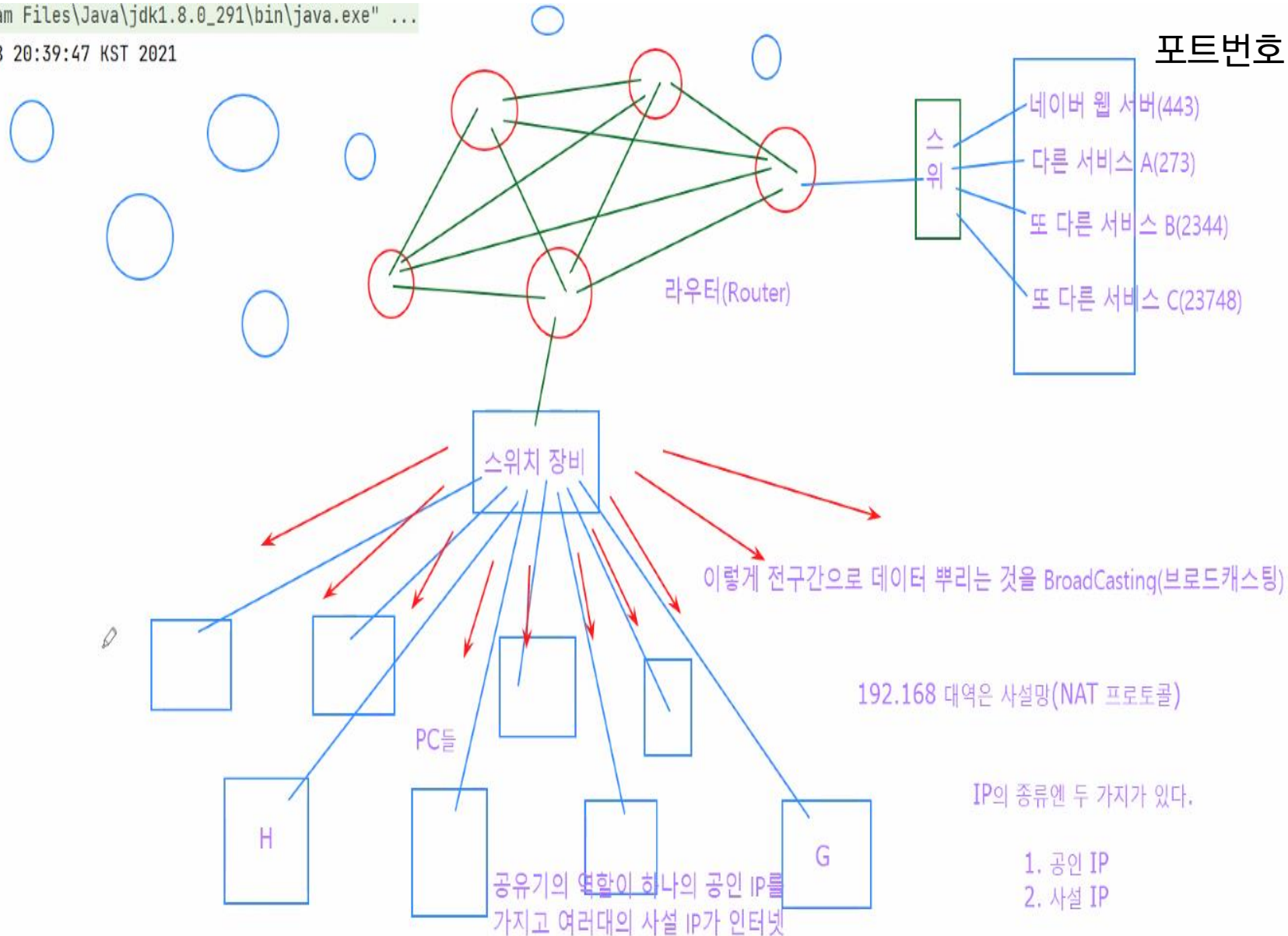
- URL을 http를 반드시 포함해서 작성

--

## ▶ 내용 : 스위치장비, 라우터, 포트번호, 공인IP

gram Files\Java\jdk1.8.0\_291\bin\java.exe" ...

08 20:39:47 KST 2021



## 주요 내용

--

인터넷을 하려면 공인 IP가 필요하다  
이걸 서비스하는 게 NAT 프로토콜이고  
NAT 프로토콜을 지원하는 게 공유기(공인 IP)

스위치 장비(공유기)는 하드웨어 주소를 보고  
어디서 작업을 하는지 판단하고  
브로드캐스팅 한다.

공유기가 사설 IP를 공인 IP로 바꿔줌.

우리가 요청하는 주소는 라우터가 판단

--

결론은 웹 서비스를 만드는데 있어  
가장 중요한 것은 **공인 IP**와 **포트번호**이다.

포트번호를 알아야 어떤 서비스로  
들어가는지 알 수 있기 때문에

## ▶ 내용 :

```
public class c_SocketServerTest {
    public static void main(String[] args) {
        // 포트의 역할 : 서비스 번호
        // 결국 우리가 어떤 서비스에 접근하기 위해서는 무엇을 알아야 한다? 아이피와 포트
        int port = Integer.parseInt(s: "33333");

        try {
            // 소켓이란?
            // 전기 분야에서 소켓에 전원 코드를 연결하면 전기 제품들이 구동 가능한것과 마찬가지로
            // 프로그래밍 분야에서 소켓이란 다른 컴퓨터와 내 컴퓨터를 연결하는 통로 역할을 한다.
            // 그러니까 통신을 수행할 수 있도록 내 소켓을 만들었음. (양쪽 모두 소켓이 있어야 통신이 가능)
            ServerSocket servSock = new ServerSocket(port);
            System.out.println("Server: Listening - " + port);

            while (true) {
                // accept() 부분에서 서버는 Blocking(블록킹) 연산을 수행하고 있음
                // accept는 스핀락의 개념. 화장실에서 안에있는 사람이 나올때까지 문을 두드린다.
                // 즉, 니가 준비될때까지 난 계속 기다린다(문을 두드리면서)

                // Blocking의 반대 개념도 있지 않을까?
                // Non-Blocking이라고 하며 비동기 처리와 관계가 깊음
                // 위의 servSock은 말그대로 서버소켓이고, 아래의 sock은 접속한 사용자의 소켓이다.
                Socket sock = servSock.accept();
                // 접속이 완료되었으면 접속한 클라이언트의 IP를 확인한다.
                System.out.println "[" + sock.getInetAddress() + "] client connected");
            }
        }
    }
}
```

## 주요 내용

```
--
Integer.parseInt(" ");
숫자형의 문자열을 인자 값으로 받으면
해당 값을 10진수 Integer 형으로
반환해준다.
--
서버소켓 객체 생성
ServerSocket servSock
                =new ServerSocket(port);
통신 수행을 위한 역할을 소켓이 한다
(양쪽에 있어야만 한다.)
--
Accept
Serversocket은 클라이언트 연결 수락을
위해 accpet를 실행하여 블로킹해야한다.
(블로킹 : 스레드가 대기상태가 된다는 의미)
---
서버 : 서비스를 제공하는 프로그램
클라이언트 : 서비스를 받는 프로그램
```



## ▶ 내용 :

```
OutputStream out = sock.getOutputStream();
// PrintWriter에 송신용 객체를 배치함으로써
// writer.println 으로 구동시키는 것이 전송되게 만들었음
PrintWriter writer = new PrintWriter(out, autoFlush: true);
// 현재 시간 정보가 클라이언트에게 전송됨
writer.println(new Date().toString());

//++ 입력이 들어올때까지 대기한다.(blocking)

// 클라이언트로부터 입력받을 객체를 생성함.(수신) InputStream
InputStream in = sock.getInputStream();
// InputStream을 사용해서 들어오는 객체는 반드시 아래와 같이 읽어야 합니다.
// InputStreamReader(): InputStream 읽기
// BufferedReader(): 데이터가 많이 들어오거나 빈번하게 지속적으로 들어올 수 있어
// 버퍼를 가진 상태에서 읽기를 지원하기 위함
BufferedReader reader = new BufferedReader(new InputStreamReader(in));
//
System.out.println("msg: " + reader.readLine());
}
} catch (IOException e) {
    System.out.println("Server Exception: " + e.getMessage());
    e.printStackTrace();
}
}
```

## 주요 내용

```
--
OutputStream /inputstream
입출력 스트림 클래스
--
BufferedReader
문자단위 입출력 스트림 클래스
--
Reader / writer
문자단위 입출력 스트림 클래스
---
IOException e
오류발생시 운영체제에서 해결
```