6월 8일 복습

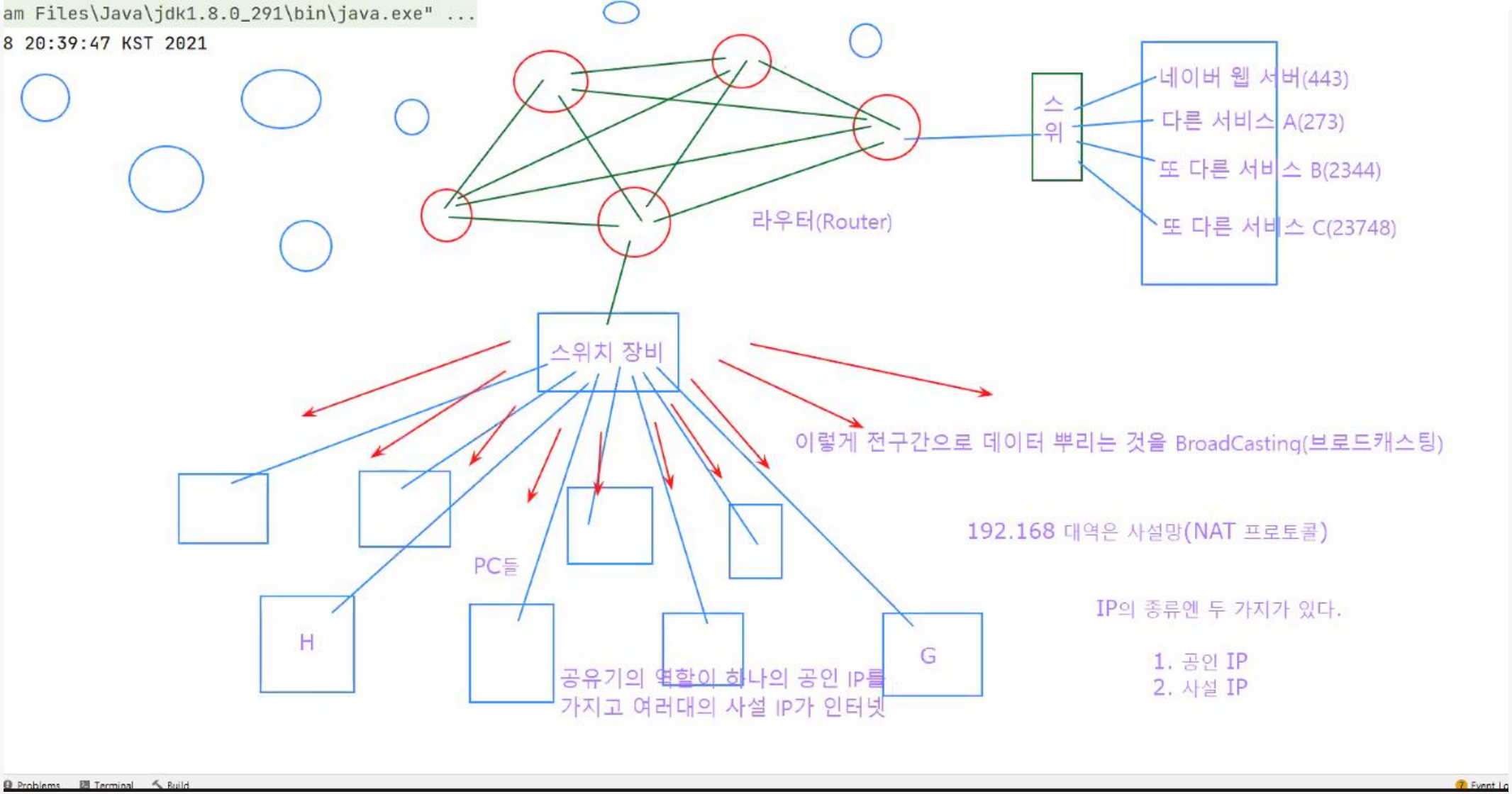
이태양


```java
public class SocketClientTest {
    public static void main(String[] args) {
        String hostname = "172.30.1.49";
        int port = 33333;

        for(int i =0; i < 10; i++){
            try {
                Socket sock = new Socket(hostname,port);
                OutputStream out = sock.getOutputStream();

                String str = "Hello Network Programming!";

                out.write(str.getBytes());

                InputStream in = sock.getInputStream();
                BufferedReader reader = new BufferedReader(new InputStreamReader(in));

                String time = reader.readLine();
                System.out.println(time);
            } catch (UnknownHostException e){
                System.out.println("Server not found" + e.getMessage());

            }catch (IOException e) {
                System.out.println("I/O Error : "+e.getMessage());
            }
        }
    }
}
```

```java
public class SocketServerTest {
    public static void main(String[] args) {
        int port = Integer.parseInt( s: "33333");

        try {
            ServerSocket servSock = new ServerSocket(port);
            System.out.println("Server ; Listening - " +port);

            while (true){
                Socket sock = servSock.accept();
                System.out.println("[" + sock.getInetAddress() + "] client connected");


                OutputStream out = sock.getOutputStream();
                PrintWriter writer = new PrintWriter(out, autoFlush: true);

                writer.println(new Date().toString());

                InputStream in = sock.getInputStream();
                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();
        }
```

포트 번호와 내 아이피를적고 통신되는지 확인해봤다

설명은 다음수업시간에!



피씨들이 다른 웹 서버에 접근하려할 때 라우터를 통해서 접근한다
라우터가 트래픽을 보고 우회도 시키고 바로 보내기도 한다


```java
class ThreadManager {
    final static int MAXTHREAD = 3;
    final static BigInteger START = new BigInteger( val: "1");
    final static BigInteger END = new BigInteger( val: "100000000000");
    final static int EVEN = 2;
    final static int SEVEN = 7;
    final static int ELEVEN = 11;
    final static int[] OPTION_ARR = { EVEN, SEVEN, ELEVEN };
    Thread[] thr;
    public ThreadManager () {
        thr = new Thread[MAXTHREAD];
        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i] = new Thread(new DistributedThread(START, END, i, OPTION_ARR[i]));
        }
    }
    public void calcEachBigInteger () {
        calcEachBigIntegerStart();
    }
    public void calcEachBigIntegerStart () {
        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i].start();
        }
    }
    public void calcEachBigIntegerJoin () throws InterruptedException {
        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i].join();
        }
    }
}
```

필요한 변수들을 선언해주고

Thr배열에 원하는값을 넣기위해 DistributedThread를 사용했다


```java
class DistributedThread implements Runnable {
    BigInteger start;
    BigInteger end;
    int threadIdx;
    int option;

    static final BigInteger ONE = new BigInteger( val: "1");

    BigInteger localSum;
    static BigInteger totalSum;

    public DistributedThread (BigInteger start, BigInteger end, int threadIdx, int option) {
        this.start = start;
        //this.end = end.add(BigInteger.ONE);
        this.end = new BigInteger( val: "30").add(BigInteger.ONE);
        this.threadIdx = threadIdx;
        this.option = option;

        localSum = BigInteger.ZERO;
        totalSum = BigInteger.ZERO;
    }

    private synchronized void addAll () {
        totalSum = totalSum.add(localSum);
    }
```

필요한 변수 선언

30번을 반복하려고 30을 넣어주고

0으로 초기화 해줌

토탈썸을 구하기위한 싱크로나이즈


```
    @Override
    public void run() {
        for (BigInteger i = start; i.compareTo(end) == -1; i = i.add(ONE)) {

            if (
                    (i.mod(new BigInteger(String.valueOf(option))).
                            compareTo(BigInteger.ZERO)
                    ) == 0
            ) {

                localSum = localSum.add(i);

            }
        }

        addAll();

        System.out.println("totalSum = " + totalSum);
    }
}

public class Prob61 {
    public static void main(String[] args) {
        ThreadManager tm = new ThreadManager();
        tm.calcEachBigInteger();
    }
}
```

compareTo()라는 비교함수를 사용해서 조건을 걸고

옵션배열의 값을 받아와 비교해주고

다 더해서 토탈썸출력