(디지털 컨버전스) 스마트 콘텐츠와 웹 융합 응용 SW개발자 양성과정

훈련기간 : 2021.05.07 ~ 2021.12.08

KH정보교육원 | 전승리

```java
import java.math.BigInteger;
class ThreadManager {
    final static int MAXTHREAD = 3; //Thread 3개 사용
    final static BigInteger START = new BigInteger( val: "1");
    final static BigInteger END = new BigInteger( val: "100000000000"); //1000억

    final static int EVEN = 2;
    final static int SEVEN = 7;
    final static int ELEVEN = 11;

    final static int[] OPTION_ARR = { EVEN, SEVEN, ELEVEN }; // 2,7,11로 나누기 조건

    Thread[] thr;

    public ThreadManager () {
        thr = new Thread[MAXTHREAD]; //3개로 초기화

        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i] = new Thread(new DistributedThread(START, END, i, OPTION_ARR[i])); //각 스레드마다 조건 조기화
                                                                                      //각 Thread마다 객체가 다르기때문에 따로 임계 영역 X
        }
    }

    public void calcEachBigInteger () throws InterruptedException { //중계 메서드 배치
        calcEachBigIntegerStart();
        calcEachBigIntegerJoin();
    }

    public void calcEachBigIntegerStart () { //Thread 시작메서드
        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i].start();
        }
    }
```

```java
    public void calcEachBigIntegerJoin () throws InterruptedException { //각 Thread가 끝날때까지 기다린다.
        for (int i = 0; i < MAXTHREAD; i++) {
            thr[i].join();
        }
    }
}
```

Quiz 스레드 문제

숫자가 굉장히 커지므로 BigInteger를 사용하도록 한다.
1 ~ 1000억까지 짝수들의 합을 계산하는 스레드 1개
1 ~ 1000억까지 7의 배수들의 합을 계산하는 스레드 1개
1 ~ 1000억까지 11의 배수를 곱하는 스레드 1개
총 3개의 스레드를 만들어서 이들 각각의 결과를 출력하고
이 결과의 합을 출력하도록 프로그래밍 해보자!

```java
class DistributedThread implements Runnable {
    BigInteger start;
    BigInteger end;

    int threadIdx; //Thread 번호
    int option;    //ThreadManager 클래스 조건 대입용

    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 = new BigInteger( val: "10").add(BigInteger.ONE);

        this.threadIdx = threadIdx;
        this.option = option;

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

    }

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

메서드에 synchronized를 사용해 함수자체에 LOCK을 건다. (this)를 확실하게 해야한다.
여기서 totalsum이 전역변수가 아니라면 메서드 내부에서 LOCK이걸려 값이 내부에서만 돌게된다.

```java
@Override
public void run() { //interface 구현

    for (BigInteger i = start; i.compareTo(end) == -1; i = i.add(ONE)) {

        /* DEBUG 메시지
        System.out.println("threadIdx: " + threadIdx +
                ", BigInteger 기반 비교를 시작합니다. 현재 i = " + i +
                ", start = " + start + ", end = " + end +
                ", i + ONE = " + i.add(ONE)); */

        // if (i % option == 0) {
        if (
                (i.mod(new BigInteger(String.valueOf(option))).
                        compareTo(BigInteger.ZERO)
                ) == 0
        ) {
            DEBUG 메시지
            System.out.println("threadIdx: " + threadIdx +
                    ", option: " + option +
                    ", 내가 찾는 숫자는: " + i);
            localSum = localSum.add(i);

            System.out.println("threadIdx: " + threadIdx + ", sum = " + localSum);
        }
    }

    addAll();

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

```
Returns: -1, 0 or 1 as this BigInteger is numerically less than, equal to, or greater than val.
public int compareTo(BigInteger val) {
```

BigIngeger.compareTo : 값을 비교해서 작으면 return -1
자바 API를 잘 찾아서 기능 사용필요

mod로 나누고 compareTo로 비교해서 같다면 return 0
int 타입이 사용불가 - > BigInteger 클래스에서 찾아서 사용

질문 : 이부분을 출력하지 않으면 왜 문제가 발생하는지 모르겠습니다.

```java
public class Prob61 {
    public static void main(String[] args) throws InterruptedException {
        ThreadManager tm = new ThreadManager();

        tm.calcEachBigInteger();
    }
}
```

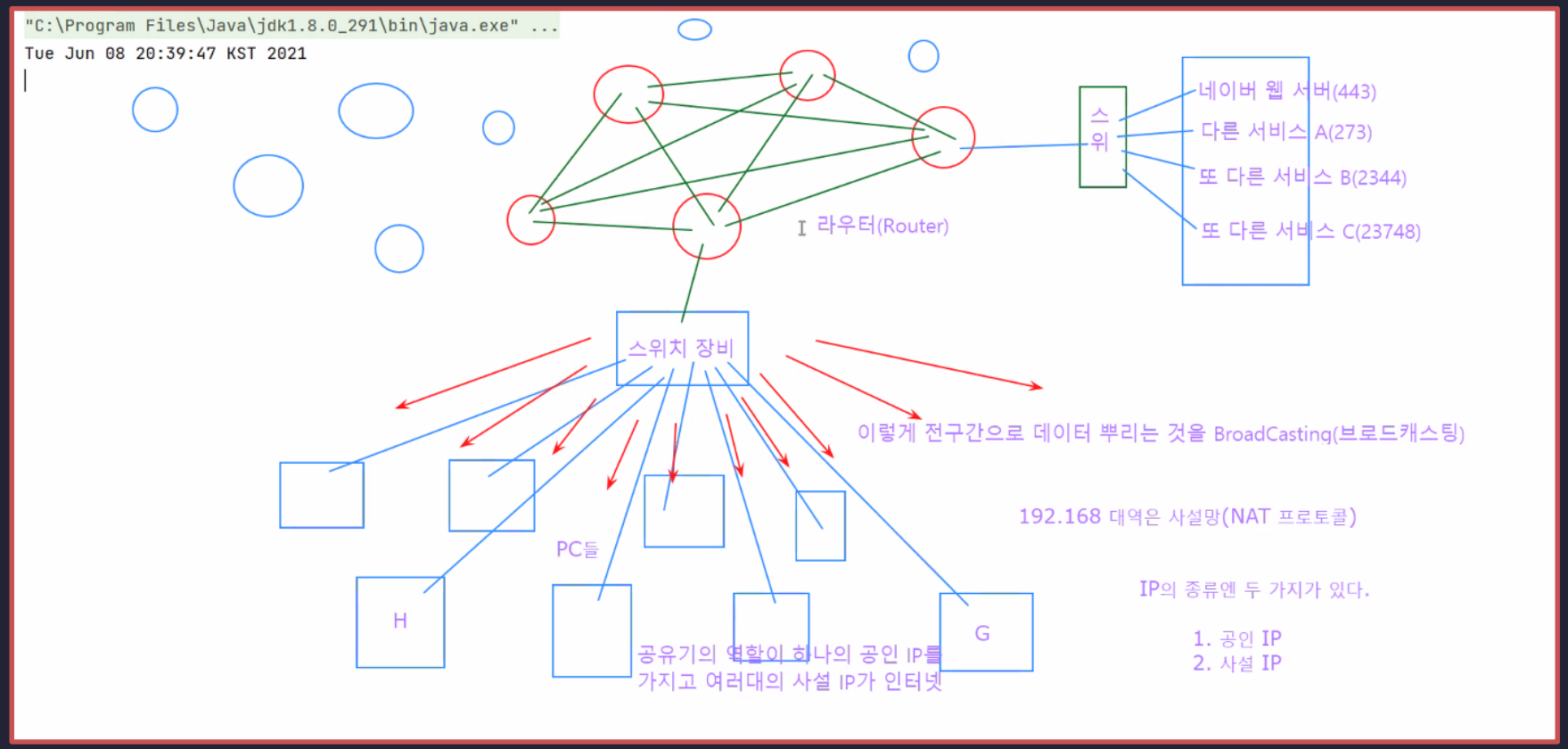
출력 O

```
totalSum = 0
threadIdx: 0, option: 2, 내가 찾는 숫자는: 2
threadIdx: 0, option: 2, 내가 찾는 숫자는: 4
threadIdx: 1, option: 7, 내가 찾는 숫자는: 7
threadIdx: 0, option: 2, 내가 찾는 숫자는: 6
threadIdx: 0, option: 2, 내가 찾는 숫자는: 8
totalSum = 7
threadIdx: 0, option: 2, 내가 찾는 숫자는: 10
totalSum = 37
```

출력 X

```
totalSum = 37
totalSum = 30
totalSum = 37
```

네트워크



스위치 장비 : 컴퓨터들을 케이블을 이용해 서로 연결해주거나 다른 허브와 연결을 통해 네트워크 사이의 연결하는 역할을 담당한다.스위치는 포트에서 입력되는 출발지 MAC 테이블에 일정시간 기억한다. 특정 MAC을 목적지로 하는 패킷이 오면 MAC 테이블을 찾아서 목적지로만 데이터를 보낸다.

라우터 : DHCP로 사설 아이피를 할당하고 사설네트워크(게이트웨이역할)를 대표한다. NAT로 사설네트워크 안에있는 기기들이 라우터를 통해 외부 네트워크와 연결될수있게 해준다.

현재는 IP와 포트번호가 가장 중요한 개념이다.
이후 수업에서 자바로 서버 클라이언트 코드에 대한 설명이 들어갈예정이다.