네트워크 프로그래밍 5주차

> 저번 주 과제
> DHCP는 ip주소와 dns 주소를 알려줌

```
package network;

import java.io.*;

public class Chap2_4_0_OutputStreamWriter {
    public static void main(String[] args) {
        try (OutputStream outFile = new
FileOutputStream("dataWriter.txt");
             OutputStreamWriter outWriter = new
OutputStreamWriter(outFile, "utf-8")) {
            outWriter.write("한밭test");
        } catch (IOException ex) {
            System.err.println(ex.getMessage());
        }        try (BufferedWriter out = new BufferedWriter(new
OutputStreamWriter(System.out, "utf-8"))) {
            out.write("한밭test");
        } catch (IOException ex) {
            System.err.println(ex.getMessage());
        }    }}
```

```
package network;

import java.io.*;

public class Chap2_3_2_DataOutputStream {
    static final String dataFile = "invoice.bin";
    static final int[] units = {12, 8, 13, 29, 50};
    static final double[] prices = {19.99, 9.99, 15.99, 3.99, 4.99};
    static final String[] descs = {"shoes", "watch", "cloth", "socks",
"notes"};

    public static void main(String[] args) throws IOException {
        try (DataOutputStream out = new DataOutputStream(new
BufferedOutputStream(new FileOutputStream(dataFile)))) {
            int totalLength = 0;
            for (int i = 0; i < prices.length; i++) {
                out.writeInt(units[i]);
                out.writeDouble(prices[i]);
//                out.writeShort(units[i]);
                out.writeUTF(descs[i]);
//                totalLength += descs[i].length();
            }
            System.out.println("string length=" + totalLength);
        }
        double total = 0.0;
        try (DataInputStream in = new DataInputStream(new
BufferedInputStream(new FileInputStream(dataFile)))) {
            double price;
            int unit;
            String desc;
            try {
                while (true) {
                    unit = in.readInt();
                    price = in.readDouble();
//                    unit=in.readShort();
                    desc = in.readUTF();
                    System.out.format("You ordered %d units of %s at
$%.2f%n", unit, desc, price);
                    total += unit * price;

                }
            } catch (EOFException e) {

            }
            System.out.format("For a TOTAL of: $%.2f%n", total);

        }
    }
}
```

Thread

TaskThread

```
package network;

public class chap3_0_0_TaskThreadDemo2 {
    public static void main(String[] args) {
        Thread printA = new PrintChar('a', 100);
        Thread printB = new PrintChar('b', 100);
        Runnable print100 = new PrintNum(100);
//        PrintNum print100 = new PrintNum(100);
        Thread thread3 = new Thread(print100);
        thread3.start();
//        thread3.run();
```

```
        printA.start();
        printB.start();
    }}

class PrintChar extends Thread {
    private char charToPrint;
    private int times;
    public PrintChar(char c, int t) {
        charToPrint = c;
        times = t;
    }
    @Override
    public void run() {
        for (int i = 0; i < times; i++) {
            System.out.print(charToPrint+" ");
        }    }}

class PrintNum implements Runnable {
    private int lastNum;

    public PrintNum(int n) {
        lastNum = n;
    }
    public void run() {
        for (int i = 1; i <= lastNum; i++) {
            System.out.print(" " + i);
        }
    }
}
```

Wait

```
package network;

class SharedResource {
    private boolean isNumberTurn = true;

    // 숫자를 출력하는 메소드
    public synchronized void printNumber(int number) {
        while (!isNumberTurn) {
            try {
                wait(); // 문자를 출력하는 스레드가 실행될 때까지 대기

            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        System.out.print(number + " ");
        isNumberTurn = false; // 다음은 문자를 출력해야 함
        notify(); // 문자 스레드에 알림

    }

// 문자를 출력하는 메소드

    public synchronized void printChar(char character) {

        while (isNumberTurn) {
            try {
                wait(); // 숫자를 출력하는 스레드가 실행될 때까지 대기
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        System.out.print(character + " ");
        isNumberTurn = true; // 다음은 숫자를 출력해야 함
        notify(); // 숫자 스레드에 알림

    }
}

class NumberThread extends Thread {
    private SharedResource resource;
    public NumberThread(SharedResource resource) {
        this.resource = resource;
    }
    public void run() {
        for (int i = 1; i <= 5; i++) {
            resource.printNumber(i); // 숫자를 출력
        }
    }
}

class CharThread extends Thread {
    private SharedResource resource;
    public CharThread(SharedResource resource) {
        this.resource = resource;
    }
    public void run() {
        for (char c = 'A'; c <= 'E'; c++) {
            resource.printChar(c); // 문자를 출력
        }
    }
}

public class chap3_0_7_Wait_New {
    public static void main(String[] args) {
        SharedResource resource = new SharedResource();
        Thread charThread = new CharThread(resource);
        Thread numberThread = new NumberThread(resource);
        numberThread.start();
        charThread.start();
    }
}
```

DigestThread

```
package network;
import java.io.*;
import java.security.DigestInputStream;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class Chap3_1_1_DigestThread extends Thread {

    private String filename;

    public Chap3_1_1_DigestThread(String filename) {
        this.filename = filename;
    }
    @Override
    public void run() {
        try {
            FileInputStream in = new FileInputStream(filename);
            MessageDigest sha = MessageDigest.getInstance("SHA-256");
            DigestInputStream din = new DigestInputStream(in, sha);
            while (din.read() != -1) ;
            din.close();
            byte[] digest = sha.digest();

            StringBuilder result = new StringBuilder(filename);
            result.append(": ");

//result.append(DatatypeConverter.printHexBinary(digest));

            result.append(byteToHex(digest));
            System.out.println(result);

        } catch (IOException ex) {
            System.err.println(ex);
        } catch (NoSuchAlgorithmException ex) {
            System.err.println(ex);
        }    }
    public static void main(String[] args) {
        String[] temp = {"data.txt", "data.bin"};
        for (String filename : temp) {
            Thread t = new Chap3_1_1_DigestThread(filename);
            t.start();
        }    }
    public static String byteToHex(byte[] bytes) {
        StringBuilder sb = new StringBuilder();
        for (byte b : bytes) {
            String st = String.format("%02X", b);
            sb.append(st);
        }        return sb.toString();
    }
}
```