

객체지향프로그래밍 12 주차 스레드

프로그래밍

```
Thread_P.java MyRunnable1.java threadMain.java DalTestMain.java Dal.java Th
1 //세명의 친구가 각자 놀다가 만나게 되는 것
2 import java.util.Random;
3
4 public class Thread_P extends Thread
5 {
6     String[] randomDoing;
7     int index;
8     Random random;
9     String name;
10
11 public Thread_P(String str, int i)
12 {
13     name=str;
14     random = new Random();
15     random.setSeed(System.currentTimeMillis());
16     random.setSeed(i);
17     randomDoing = new String[] {"TV보기", "산책하기", "자마시기", "요리하기", "책읽기"};
18
19     //random.nextInt(5);
20
21 }
22
23 public void run()
24 {
25     for(int i=0; i<10; i++)
26     {
27         System.out.println((10-i) + "시간 전");
28         index = random.nextInt(5);
29         if(index == 0)
30             System.out.println(name + "가 지금 하는 일 : " + randomDoing[0]);
31         else if(index==1)
32             System.out.println(name + "가 지금 하는 일 : " + randomDoing[1]);
33         else if(index==2)
34             System.out.println(name + "가 지금 하는 일 : " + randomDoing[2]);
35         else if(index==3)
36             System.out.println(name + "가 지금 하는 일 : " + randomDoing[3]);
37         else if(index==4)
38             System.out.println(name + "가 지금 하는 일 : " + randomDoing[4]);
39         //
40         // System.out.print("이상한 값이 나올");
41         try {
42             sleep(1500);
43         } catch (Exception e)
44         {}
45     }
46 }
47 }
48 }
49 }
```

네 사람이 같이 게임을 하기로 했는데 게임을 하기 전에 무슨 일을 했을지 알려주는 프로그램을 만들었습니다.

할 수 있는 행동은 TV 보기, 산책하기, 차마시기, 요리하기, 책읽기가 있고, 1.5 초마다 하는 일이 바뀔 수도, 안 바뀔 수도 있습니다.

네 사람이 각자 행동을 하다가 시간이 되면 게임을 하는 프로그램입니다.

5 가지 활동을 배열로 만든 후

랜덤 객체를 이용하여 5 가지 활동 중 한가지 활동을 랜덤으로 선택하도록 했습니다.

쓰레드가 실행되고 1.5 초마다 한번씩 행동을 바꾸거나 안바꾸거나(동일한 랜덤수가 나오면 행동이 바뀌지 않습니다)하도록 하는 Person 을 의미하는 P 를 추가한 Thread_P 클래스입니다.

```
Thread_P.java MyRunnable1.java threadMain.java DalTestMain.java
1
2 public class threadMain extends Thread{
3 public static void main(String args[]) throws InterruptedException
4 {
5     System.out.println("현서와 주비, 비니, 그리고 고파가 같이 게임하기로 했다.\n그들은 만나기 전에 무슨 일을 했을까?");
6     sleep(500);
7     Thread_P person1 = new Thread_P("현서", 1);
8     Thread_P person2 = new Thread_P("주비", 2);
9     Thread_P person3 = new Thread_P("비니", 3);
10    Thread_P person4 = new Thread_P("고파", 3);
11    Thread_T timer = new Thread_T();
12    // sleep(1000);
13    person1.start();
14    // sleep(1000);
15    person2.start();
16    // sleep(1000);
17    person3.start();
18    person4.start();
19    timer.start();
20    // sleep(800);
21    // person1.run();
22
23 }
24 }
25 }
```

Main 클래스로 네 사람 객체와 타이머 객체를 만든 후 쓰레드를 실행시킵니다.

상황설명을 한 후 각자의 행동이 출력되고, 시간이 되면 모두 게임을 하는 출력을 합니다.

```

1 public class Thread_T extends Thread
2 {
3     Thread_T()
4     {
5     }
6 }
7
8 public void run()
9 {
10     try {
11         sleep(8000);
12     } catch (InterruptedException e) {
13     }
14     System.out.println("이제 만나러갈 시간입니다");
15     System.out.println("형서, 준비, 비니, 고파는 지금 게임을 하고 있어요");
16     System.exit(0);
17 }
18 }

```

타이머 클래스로 Timer 의 T 를 따 Thread_T 라는 이름의 클래스를 만들었습니다.

사람들이 어떤 행동을 하고 있는지 알 수 없지만 start 가 된 후 8 초 후에 모두 행동을 종료하고 게임을 합니다.

```

<terminated> threadMain [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2021. 5. 29. 오후 11:07:25 - 오후 11:07:37)
이제 만나러갈 시간입니다.
그들은 만나기 전에 무슨 일을 했을까?
준비가 지금 하는 일 : 요리하기
형서가 지금 하는 일 : TV보기
비니가 지금 하는 일 : 책읽기
고파가 지금 하는 일 : 책읽기
비니가 지금 하는 일 : TV보기
형서가 지금 하는 일 : 요리하기
준비가 지금 하는 일 : 자마시기
고파가 지금 하는 일 : TV보기
비니가 지금 하는 일 : TV보기
준비가 지금 하는 일 : TV보기
형서가 지금 하는 일 : 자마시기
고파가 지금 하는 일 : TV보기
비니가 지금 하는 일 : 산책하기
준비가 지금 하는 일 : 자마시기
형서가 지금 하는 일 : 요리하기
고파가 지금 하는 일 : 산책하기
비니가 지금 하는 일 : 요리하기
준비가 지금 하는 일 : 책읽기
형서가 지금 하는 일 : 책읽기
고파가 지금 하는 일 : 요리하기
비니가 지금 하는 일 : 자마시기
형서가 지금 하는 일 : 책읽기
준비가 지금 하는 일 : TV보기
고파가 지금 하는 일 : 자마시기
이제 만나러갈 시간입니다
형서, 준비, 비니, 고파는 지금 게임을 하고 있어요

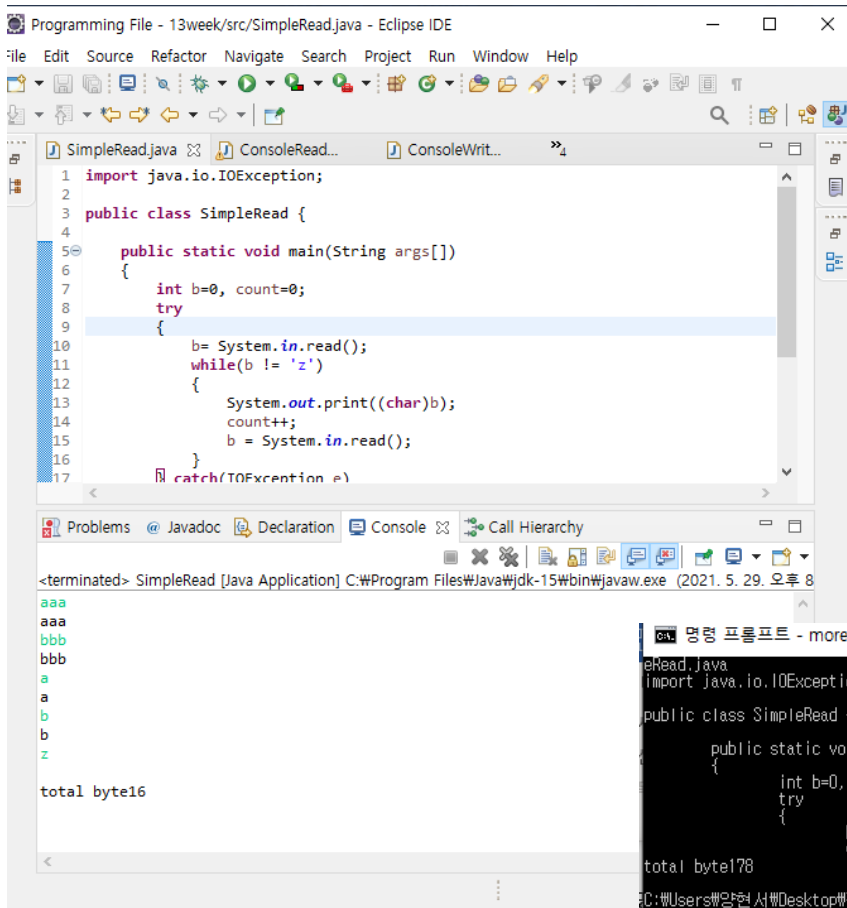
```

실제로 출력되는 모습입니다.

Sleep 이 되면서 출력이 되고, for 문을 모두 반복하진 않았지만 정해진 시간이 되면 모두 모여 게임을 합니다.

객체지향프로그래밍 13 주차 정리

프로그래밍



```
1 import java.io.IOException;
2
3 public class SimpleRead {
4
5     public static void main(String args[])
6     {
7         int b=0, count=0;
8         try
9         {
10             b= System.in.read();
11             while(b != 'z')
12             {
13                 System.out.print((char)b);
14                 count++;
15                 b = System.in.read();
16             }
17         } catch (IOException e)
18         {
19
20         }
21     }
22 }
```

Console Output:

```
<terminated> SimpleRead [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2021. 5. 29. 오후 8
aaa
aaa
bbb
bbb
a
a
b
b
z
total byte16
```


IO 실습

z 가 나올 때까지 반복해서 값을
입력받고, z 가 입력되면 total
byte 값을 출력한다

IO 실습

Cmd 창에서 위의 프로그램 실행한
결과

output .txt 파일에 결과 저장 및
파일 내용 실행



```
C:\Users\양현서\Desktop\객체\Programming File\13week\src>java SimpleRead < SimpleRead.java > output.txt
지정된 파일을 찾을 수 없습니다.

C:\Users\양현서\Desktop\객체\Programming File\13week\src>java SimpleRead < SimpleRead.java > output .txt
import java.io.IOException;

public class SimpleRead {

    public static void main(String args[])
    {
        int b=0, count=0;
        try
        {
            b= System.in.read();
            while(b != 'z')
            {
                System.out.print((char)b);
                count++;
                b = System.in.read();
            }
        } catch (IOException e)
        {

        }
    }
}

total byte178
C:\Users\양현서\Desktop\객체\Programming File\13week\src>.txt 파일을 액세스할 수 없습니다.

C:\Users\양현서\Desktop\객체\Programming File\13week\src>more output .txt
import java.io.IOException;

public class SimpleRead {

    public static void main(String args[])
    {
        int b=0, count=0;
        try
        {
            b= System.in.read();
            while(b != 'z')
            {
                System.out.print((char)b);
                count++;
                b = System.in.read();
            }
        } catch (IOException e)
        {

        }
    }
}

total byte178
```

Programming File - 13week/src/ConsoleReaderMain.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

```
SimpleRead.java ConsoleReader.java ConsoleReaderMain.java
13 {
14     ConsoleReader in = new ConsoleReader(System.in);
15     System.out.print("정수를 입력해주세요 : ");
16     ivalue = in.readInt();
17     System.out.println("value = "+ivalue);
18     System.out.print("\n실수를 입력해주세요 : ");
19     dvalue = in.readDouble();
20     System.out.println("vlaue = " + dvalue);
21     System.out.print("\n문자를 입력해주세요 : ");
22     cvalue = in.readChar();
23     System.out.println("value = " + cvalue);
24     System.out.print("\n문자열을 입력해주세요 : ");
25     svalue = in.readString();
26     System.out.println("value = "+svalue);
27 } catch (Exception e)
28 {
29     System.out.println("타입에 맞지 않은 입력입니다");
30 }
```

Problems Javadoc Declaration Console Call Hierarchy

<terminated> ConsoleReaderMain [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2021. 5. 29)

정수를 입력해주세요 : 10
value = 10

실수를 입력해주세요 : 10.1
vlaue = 10.1

문자를 입력해주세요 : Y
value = Y

문자열을 입력해주세요 : YangHyeonSeo
value = YangHyeonSeo

Writable Smart Insert

IO 실습

원하는 타입의 값을 입력받는 프로그램

ConsoleReader.java 의 함수를 이용하여 값을 입력 받는다.

IO 실습

ConsoleWriter.java 파일 안에서 함수를 정의, 실행한 함수에 따라 출력 포맷이 다르다

Programming File - 13week/src/ConsoleWriterMain.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

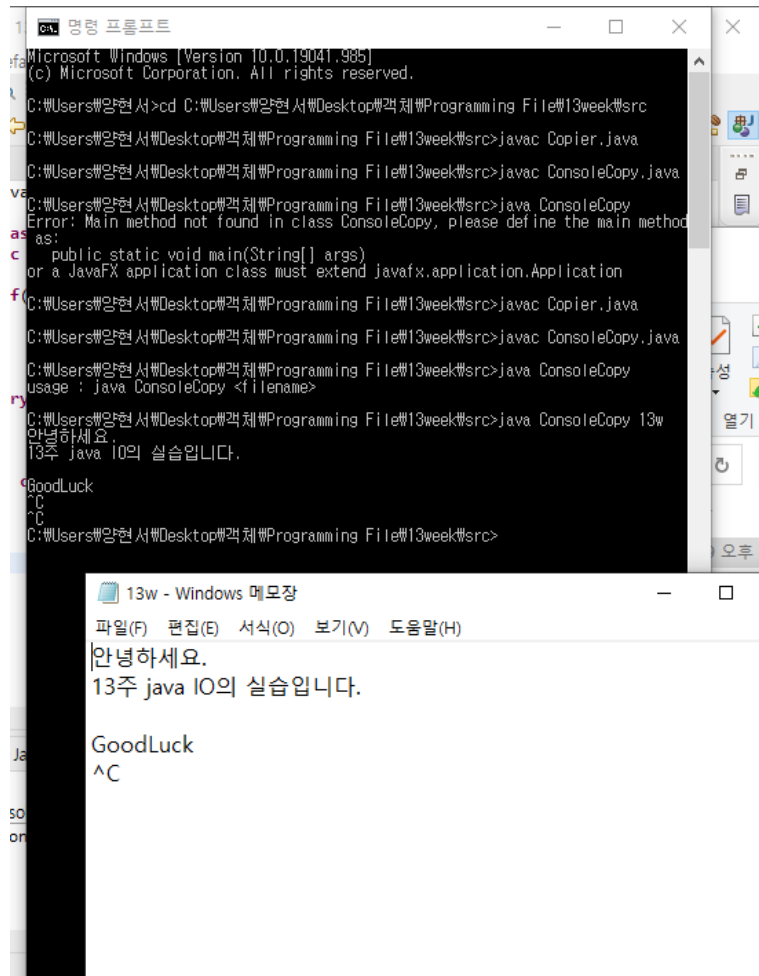
```
ConsoleRead... ConsoleRead... ConsoleWrit... ConsoleWrit...
1 public class ConsoleWriterMain {
2     public static void main(String args [])
3     {
4         try
5         {
6             ConsoleWriter out = new ConsoleWriter(System.out);
7             double d = 123.456789;
8             int money = 1234567;
9
10            out.println(d);
11            out.println(d, 2);
12            out.printlnWon(money);
13            out.printlnDollar(money);
14        } catch (Exception e)
15        {
16            System.out.println(e);
17        }
18    }
19 }
20 }
21 }
22 }
```

Problems Javadoc Declaration Console Call Hierarchy

<terminated> ConsoleWriterMain [Java Application] C:\Program Files\Java\jdk-15\bin\javaw.exe (2021. 5. 29)

123.456789
123.46
₩1,234,567
\$1,234,567.00

Writable Smart Insert



```
1 CA 명령 프롬프트
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. All rights reserved.

C:\Users\양현서>cd C:\Users\양현서\Desktop\객체\Programming File\13week\src
C:\Users\양현서\Desktop\객체\Programming File\13week\src>javac Copier.java
C:\Users\양현서\Desktop\객체\Programming File\13week\src>javac ConsoleCopy.java
C:\Users\양현서\Desktop\객체\Programming File\13week\src>java ConsoleCopy
Error: Main method not found in class ConsoleCopy, please define the main method
as:
    public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
C:\Users\양현서\Desktop\객체\Programming File\13week\src>javac Copier.java
C:\Users\양현서\Desktop\객체\Programming File\13week\src>javac ConsoleCopy.java
C:\Users\양현서\Desktop\객체\Programming File\13week\src>java ConsoleCopy
usage: java ConsoleCopy <filename>
C:\Users\양현서\Desktop\객체\Programming File\13week\src>java ConsoleCopy 13w
안녕하세요.
13주 Java IO의 실습입니다.

GoodLuck
^C
C:\Users\양현서\Desktop\객체\Programming File\13week\src>
```

13w - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

안녕하세요.
13주 java IO의 실습입니다.

GoodLuck
^C

IO 실습

Cmd 창에서 ConsoleCopy.java
프로그램 실행한 결과

Java ConsoleCopy 13w 을 함으로
새로운 파일을 생성 및 입력

IO

Output : 실행 결과를 어떻게 보여줄 것인가

Input : 무엇을 읽을 것인지

⇒ 프로그램을 실행하면서 꼭 필요하게 된다

키보드로 입력 : Standard input

화면에 출력 : Standard output

키보드와 화면 이외의 입력, 출력은 어떤 것일까

HDD(하드디스크) : File input / File output : 파일의 형태로 input, output 이 이뤄진다

다른 디바이스 : 하드디스크 외에 부착된 디바이스로부터 input, output

Ex) 카메라, 마이크 등

*Input 디바이스와 Output 디바이스는 다르다

(화면, 하드디스크, 다른 디바이스 등 여러 방법으로 출력이 가능하다)

메모리 : 속도를 빠르게 하기 위해 메인 메모리의 특정 공간의 정보를 input, 이 공간에 정보를 output. 속도가 빠르고, 빅데이터, AI 등에서 많이 사용한다

네트워크프로그램 NIC(Network Interface Card) : 모든 기기가 연결되어 있음 -> 혼자 동작할 수 있는 기기에 다른 기기에 있는 데이터를 가져와야 한다. NIC 가 부착된 디바이스는 NIC 를 통해 다른 디바이스로부터 데이터를 받을 수 있고, 다른 디바이스에 보낼 수도 있다

Network Input / Network Output

→ Scket Input / Scket Output
(Scket Program)

⇒ IO 프로그램이 필요한 이유 : File input/output

Socket input/output

Standard input/output(키보드 input, 화면 output)

Input Stream

Output Stream

⇒ IO 프로그램, Socket 프로그램 모두 지원

소켓프로그램의 기본 : 자바 IO

소켓을 만들 -> inputStream : 데이터를 받아들인다 / 키보드, 파일 등이 input 장치

Inputsocket : 소켓이 input 장치

outputStream

⇒ InputStream, InputSocket, OutputStream 모두 방법이 똑같다

★왜 IO 를 써야 하는가

메인 메모리에서 프로그램이 실행될 때 데이터를 누구에게 받아야하고 누구에게 줘야만 한다

데이터를 주는 상대, 데이터를 받는 상대가 누구인지에 따라 input/output 의 형태가 다르고,
데이터의 위치, 전달 방법도 다르다

IO stream

Stream : 실행된 데이터에 대한 흐름

Input Sources : Standard in, HDD, Connected Devices with NIC



Stream

Output Sources : Standard out, HDD, Connected Devices with NIC

Byte IO

: 일반적인 데이터 전송

Ex)파일 IO(비디오, 사진 등)

- InputStream
- outputStream

Text IO

: 텍스트단위의 데이터 전송

Ex)캐릭터 단위, 문자/카톡 같은 경우

- Reader

➤ Writer

(Node Stream) 데이터 생성소스 -> 필터링(IO 데이터 형태에 따라 Byte / Text IO) -> 프로그램

NodeStream Class

Top-level : **InputStream, OutputStream** / Reader, Writer

Standard I/O(콘솔) : System.in, System.out / Reader, Writer

Files(디스크) : **FileInputStream, FileOutputStream** / FileReader, FileWriter

Top-Level(NIC) : InputStream, OutputStream

Memory : ByteArrayInputStream, ByteArrayOutputStream / CharArrayReader, CharArrayWriter

필터링 Class

Buffering : **BufferedInputStream, BufferedOutputStream / BufferedReader, BufferedWriter**

Filtering : FilterInputStream, FilterOutputStream / FilterReader, FilterWriter

Data Processing : DataInputStream, DataOutputStream

Printing : PrintStream / PrintWriter

Transferring Byte to Text : / InputStreamReader, OutputStreamWriter (Byte 를 Text 로 변환)

IO 이용

가장 많이 쓰는 방법

입력장치(NodeClass) -> **Buffering**

(자바프로그램)

Buffring -> 출력장치

변환이 들어가는 경우(ex)문자 프로그램)

입력장치(NodeClass) -> 변환(Byte to Text) -> **Buffering**

(자바프로그램)

변환 -> Bufferring->출력장치