

Object – Oriented Programming

LAB #8-2. File I/O

Stream

- 자료 입출력을 도와주는 중간 매개체
- 하나의 스트림은 단 방향 통신만 가능하다.
- 입력 Stream : 데이터 소스로부터 데이터를 읽기 위한 Stream
- 출력 Stream : 데이터 소스에 데이터를 쓰기 위한 Stream

Scanner를 사용한 File Input

Display 10.3 Reading Input from a Text File Using Scanner

```
1  import java.util.Scanner;
2  import java.io.FileInputStream;
3  import java.io.FileNotFoundException;
4
5  public class TextFileScannerDemo
6  {
7      public static void main(String[] args)
8      {
9          System.out.println("I will read three numbers and a line");
10         System.out.println("of text from the file morestuff.txt.");
11
12         Scanner inputStream = null;
13
14         try
15         {
16             inputStream =
17                 new Scanner(new FileInputStream("morestuff.txt"));
18         }
```

(continued)

Scanner를 사용한 File Input

Display 10.3 Reading Input from a Text File Using Scanner

```
19      catch(FileNotFoundException e) { 읽으려는 파일이 존재하지 않을 시 처리
20      {
21          System.out.println("File morestuff.txt was not found");
22          System.out.println("or could not be opened.");
23          System.exit(0);
24      }
25      int n1 = inputStream.nextInt( );
26      int n2 = inputStream.nextInt( );
27      int n3 = inputStream.nextInt( );
28
29      inputStream.nextLine(); //To go to the next line
30
31      String line = inputStream.nextLine( );
32
```

여기쓰기엔 구분선 생략 입력 받음

(continued)

Scanner를 사용한 File Input

Display 10.3 Reading Input from a Text File Using Scanner

```
33         System.out.println("The three numbers read from the file are:");
34         System.out.println(n1 + ", " + n2 + ", and " + n3);
35
36         System.out.println("The line read from the file is:");
37         System.out.println(line);
38
39         inputStream.close( );
40     }
41 }
```

File morestuff.txt

```
1 2
3 4
Eat my shorts.
```

This file could have been made with a text editor or by another Java program.

(continued)

Scanner를 사용한 File Input

Display 10.3 Reading Input from a Text File Using Scanner

SCREEN OUTPUT

```
I will read three numbers and a line  
of text from the file morestuff.txt.  
The three numbers read from the file are:  
1, 2, and 3  
The line read from the file is:  
Eat my shorts.
```

Scanner를 사용한 File Input / Output

Display 10.4 Checking for the End of a Text File with hasNextLine

```
1  import java.util.Scanner;
2  import java.io.FileInputStream;
3  import java.io.FileNotFoundException;
4  import java.io.PrintWriter;
5  import java.io.FileOutputStream;
6
7  public class HasNextLineDemo
8  {
9      public static void main(String[] args)
10     {
11         Scanner inputStream = null;
12         PrintWriter outputStream = null;
```

(continued)

Scanner를 사용한 File Input / Output

Display 10.4 Checking for the End of a Text File with hasNextLine

```
13      try
14      {
15          inputStream =
16              new Scanner(new FileInputStream("original.txt"));
17          outputStream = new PrintWriter(
18                      new FileOutputStream("numbered.txt"));
19      }
20      catch(FileNotFoundException e) { 읽으려는 파일이 존재하지 않을 시 처리
21      {
22          System.out.println("Problem opening files.");
23          System.exit(0);
24      }

25      String line = null;
26      int count = 0;
```

(continued)

Scanner를 사용한 File Input / Output

Display 10.4 Checking for the End of a Text File with hasNextLine

```
27      while (inputStream.hasNextLine( ))
28      {
29          line = inputStream.nextLine( );
30          count++;
31          outputStream.println(count + " " + line);
32      }

33      inputStream.close( );
34      outputStream.close( );
35  }

36 }
```

(continued)

Scanner를 사용한 File Input / Output

Display 10.4 Checking for the End of a Text File with hasNextLine

File original.txt

```
Little Miss Muffet  
sat on a tuffet  
eating her curves away.  
Along came a spider  
who sat down beside her  
and said "Will you marry me?"
```

File numbered.txt (after the program is run)

```
1 Little Miss Muffet  
2 sat on a tuffet  
3 eating her curves away.  
4 Along came a spider  
5 who sat down beside her  
6 and said "Will you marry me?"
```

BufferedReader를 이용한 Input

Display 10.7 Reading Input from a Text File Using BufferedReader

```
1  import java.io.BufferedReader;
2  import java.io.FileReader;
3  import java.io.FileNotFoundException;
4  import java.io.IOException;

5  public class TextFileInputDemo
6  {
7      public static void main(String[] args)
8      {
9          try
10         {
11             BufferedReader inputStream =
12                 new BufferedReader(new FileReader("morestuff2.txt"));

13             String line = inputStream.readLine();
14             System.out.println(
15                 "The first line read from the file is:");
16             System.out.println(line);
```

(continued)

BufferedReader를 이용한 Input

Display 10.7 Reading Input from a Text File Using BufferedReader

```
17
18     line = inputStream.readLine();
19     System.out.println(
20         "The second line read from the file is:");
21     System.out.println(line);
22     inputStream.close();
23 }
24 catch(FileNotFoundException e) 읽으려는 파일이 존재하지 않을 시 처리
25 {
26     System.out.println("File morestuff2.txt was not found");
27     System.out.println("or could not be opened.");
28 }
29 catch(IOException e) 모든 읽기/쓰기 오류
30 {
31     System.out.println("Error reading from morestuff2.txt.");
32 }
33 }
34 }
```

(continued)

BufferedReader를 이용한 Input

Display 10.7 Reading Input from a Text File Using BufferedReader

File morestuff2.txt

```
1 2 3  
Jack jump over  
the candle stick.
```

*This file could have been made with a
text editor or by another Java
program.*

SCREEN OUTPUT

The first line read from the file is:

```
1 2 3
```

The second line read from the file is:

```
Jack jump over
```

File 클래스를 이용한 txt 생성

```
import java.util.Scanner;
import java.io.File;
import java.io.PrintWriter;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;

public class FileClassDemo
{
    public static void main(String[] args)
    {
        Scanner keyboard = new Scanner(System.in);
        String line = null;
        String fileName = null;

        System.out.println("I will store a line of text for you.");
        System.out.println("Enter the line of text:");
        line = keyboard.nextLine();
    }
}
```


File 클래스를 이용한 txt 생성

```
System.out.println("Enter a file name to hold the line:");
fileName = keyboard.nextLine();
File fileObject = new File(fileName);

while (fileObject.exists())
{
    System.out.println("There already is a file named "
        + fileName);
    System.out.println("Enter a different file name:");
    fileName = keyboard.nextLine();
    fileObject = new File(fileName);
}

PrintWriter outputStream = null; If you wish, you can use fileObject,
    instead of fileName as the arg
try
{
    outputStream =
        new PrintWriter(new FileOutputStream(fileName));
}
catch (FileNotFoundException e) 읽으려는 파일이 존재하지 않을 시 처리
{
    System.out.println("Error opening the file " + fileName);
    System.exit(0);
}

System.out.println("Writing \"" + line + "\"");
System.out.println("to the file " + fileName);
outputStream.println(line);

outputStream.close();
System.out.println("Writing completed.");
}
```

Sample Dialogue

myLine.txt but that there is n

I will store a line of text for you.
Enter the line of text:
May the hair on your toes grow long and curly.
Enter a file name to hold the line:
myLine.txt
There already is a file named myLine.txt
Enter a different file name:
mySaying.txt
Writing "May the hair on your toes grow long and curly."
to the file mySaying.txt
Writing completed.