# Chapter 15

입출력

I/O

# [ 연습문제 ]

**[15-1]** 커맨드라인으로 부터 파일명과 숫자를 입력받아서, 입력받은 파일의 내용의 처음부터 입력받은 숫자만큼의 라인을 출력하는 프로그램(FileHead.java)을 작성하라.

**[Hint]** BufferedReader의 readLine()을 사용하라.

**[실행결과]**

```
C:\jdk1.8\work\ch15>java FileHead 10
USAGE: java FileHead 10 FILENAME

C:\jdk1.8\work\ch15>java FileHead 10 aaa
aaa은/는 디렉토리이거나, 존재하지 않는 파일입니다.

C:\jdk1.8\work\ch15>java FileHead 10 FileHead.java
1:import java.io.*;
2:
3:class FileHead
4:{
5:     public static void main(String[] args)
6:     {
7:            try {
8:                   int line = Integer.parseInt(args[0]);
9:                   String fileName = args[1];
10:

C:\jdk1.8\work\ch15>
```

**[15-2]** 지정된 이진파일의 내용을 실행결과와 같이 16진수로 보여주는 프로그램(HexaViewer.java)을 작성하라.

**[Hint]** PrintStream과 printf()를 사용하라.

**[실행결과]**

```
C:\jdk1.8\work\ch15>java HexaViewer HexaViewer.class
CA FE BA BE 00 00 00 31 00 44 0A 00 0C 00 1E 09
00 1F 00 20 08 00 21 0A 00 08 00 22 0A 00 1F 00
23 07 00 24 0A 00 06 00 25 07 00 26 0A 00 08 00
27 0A 00 06 00 28 08 00 29 07 00 2A 0A 00 2B 00
2C 0A 00 08 00 2D 0A 00 08 00 2E 0A 00 06 00 2F
0A 00 08 00 2F 07 00 30 0A 00 12 00 31 07 00 32
01 00 06 3C 69 6E 69 74 3E 01 00 03 28 29 56 01
00 04 43 6F 64 65 01 00 0F 4C 69 6E 65 4E 75 6D
62 65 72 54 61 62 6C 65 01 00 04 6D 61 69 6E 01
00 16 28 5B 4C 6A 61 76 61 2F 6C 61 6E 67 2F 53
... 중간생략

C:\jdk1.8\work\ch15>
```

**[15-3]** 다음은 디렉토리의 요약정보를 보여주는 프로그램이다. 파일의 개수, 디렉토리의 개수, 파일의 총 크기를 계산하는 countFiles()를 완성하시오.

**[연습문제]**/ch15/DirectoryInfoTest.java

```
import java.io.*;

class DirectoryInfoTest {
   static int totalFiles = 0;
   static int totalDirs = 0;
   static int totalSize = 0;

   public static void main(String[] args) {
        if(args.length != 1) {
            System.out.println("USAGE : java DirectoryInfoTest DIRECTORY");
            System.exit(0);
        }

        File dir = new File(args[0]);

        if(!dir.exists() || !dir.isDirectory()) {
            System.out.println("유효하지 않은 디렉토리입니다.");
            System.exit(0);
        }

        countFiles(dir);

        System.out.println();
        System.out.println("총 " + totalFiles + "개의 파일");
        System.out.println("총 " + totalDirs + "개의 디렉토리");
        System.out.println("크기 " + totalSize + " bytes");
   } // main

   public static void countFiles(File dir) {
        /*
          (1) 아래의 로직에 맞게 코드를 작성하시오.
           1. dir의 파일목록(File[])을 얻어온다.
           2. 얻어온 파일목록의 파일 중에서...
              디렉토리이면, totalDirs의 값을 증가시키고 countFiles()를 재귀호출한다.
           3. 파일이면, totalFiles를 증가시키고 파일의 크기를 totalSize에 더한다.
        */
   } // countFiles
}
```

**[실행결과]**

```
C:\jdk1.8\work\ch15>java DirectoryInfoTest .

총 786개의 파일
총 27개의 디렉토리
크기 2566228 bytes

C:\jdk1.8\work\ch15>
```

**[15-4]** 커맨드라인으로 부터 여러 파일의 이름을 입력받고, 이 파일들을 순서대로 합쳐서 새로운 파일을 만들어 내는 프로그램(FileMergeTest.java)을 작성하시오. 단, 합칠 파일의 개수에는 제한을 두지 않는다.

**[실행결과]**

```
C:\jdk1.8\work\ch15>java FileMergeTest
USAGE: java FileMergeTest MERGE_FILENAME FILENAME1 FILENAME2 ...

C:\jdk1.8\work\ch15>java FileMergeTest result.txt 1.txt 2.txt 3.txt

C:\jdk1.8\work\ch15>type result.txt
1111111111
2222222222
33333333333333

C:\jdk1.8\work\ch15>java FileMergeTest result.txt 1.txt 2.txt

C:\jdk1.8\work\ch15>type result.txt
1111111111
2222222222

C:\jdk1.8\work\ch15>type 1.txt
1111111111

C:\jdk1.8\work\ch15>type 2.txt
2222222222

C:\jdk1.8\work\ch15>type 3.txt
33333333333333

C:\jdk1.8\work\ch15>
```

**[15-5]** 다음은 FilterWriter를 상속받아서 직접 구현한 HtmlTagFilterWriter를 사용해서 주어진 파일에 있는 태그를 모두 제거하는 프로그램이다. HtmlTagFilterWriter의 write()가 태그를 제거하도록 코드를 완성하시오.

**[연습문제]**/ch15/Exercise15_5.java

```
import java.io.*;

class Exercise15_5
{
   public static void main(String[] args)
   {
       if(args.length != 2) {
           System.out.println("USAGE:    java    Exercise15_5    TAGET_FILE
RESULT_FILE");
           System.exit(0);
       }

       String inputFile  = args[0];
       String outputFile = args[1];

       try {
           BufferedReader input
                              = new BufferedReader(new FileReader(inputFile));
           HtmlTagFilterWriter output
                         = new HtmlTagFilterWriter(new FileWriter(outputFile));

           int ch = 0;

           while((ch=input.read())!=-1) {
               output.write(ch);
           }

           input.close();
           output.close();

       } catch(IOException e) {}
   }
}


class HtmlTagFilterWriter extends FilterWriter {
   StringWriter tmp = new StringWriter();
   boolean inTag = false;

   HtmlTagFilterWriter(Writer out) {
       super(out);
   }

   public void write(int c) throws IOException {
       /*
         (1) 아래의 로직에 맞게 코드를 작성하시오.
         1. 출력할 문자(c)가 '<'이면 inTag의 값을 true로 한다.
         2. 출력할 문자(c)가 '>'이면 inTag의 값을 false로 한다.
              새로운 StringWriter를 생성한다.(기존 StringWriter의 내용을 버린다.)
         3. inTag의 값이 true이면, StringWriter에 문자(c)를 출력하고
```

```
            inTag의 값이 false이면, out에 문자(c)를 출력한다.

          [참고] 태그가 시작되면 StringWriter에 출력하고 태그가 끝나면
                 StringWriter는 비워진다.
        */
    }

    public void close() throws IOException {
        out.write(tmp.toString()); // StringWriter의 내용을 출력하고
        super.close();          // 조상의 close()를 호출해서 기반 스트림을 닫는다.
    }
}
```

**[실행결과]**

```
C:\jdk1.8\work\ch15>java Exercise15_5 test.html result.txt

C:\jdk1.8\work\ch15>type result.txt



 New Document


> 안녕하세요. 태그 없애기 테스트용 파일입니다.

< 태그가 열린 채로 끝난 것은 태그로 처리하지 마세요.

C:\jdk1.8\work\ch15>type test.html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
<HEAD>
<TITLE> New Document </TITLE>
</HEAD>
<BODY>
> 안녕하세요. 태그 없애기 테스트용 파일입니다.
</BODY>
< 태그가 열린 채로 끝난 것은 태그로 처리하지 마세요.
```

**[15-6]** 다음은 콘솔 명령어 중에서 디렉토리를 변경하는 cd명령을 구현한 것이다. 알맞은 코드를 넣어 cd()를 완성하시오.

**[연습문제]**/ch15/Exercise15_6.java

```
import java.io.*;
import java.util.*;
import java.util.regex.*;

class Exercise15_6 {
   static String[] argArr;     // 입력한 매개변수를 담기위한 문자열배열
   static File curDir;         // 현재 디렉토리

   static {
       try {
           curDir = new File(System.getProperty("user.dir"));
       } catch(Exception e) {}
   }

   public static void main(String[] args) {
       Scanner s = new Scanner(System.in);

       while(true) {
           try {
               String prompt = curDir.getCanonicalPath() + ">>";
               System.out.print(prompt);

               // 화면으로부터 라인단위로 입력받는다.
               String input = s.nextLine();

               input = input.trim(); // 입력받은 값에서 불필요한 앞뒤 공백을 제거한다.
               argArr = input.split(" +");

               String command = argArr[0].trim();

               if("".equals(command)) continue;

               command = command.toLowerCase(); // 명령어를 소문자로 바꾼다.

               if(command.equals("q")) { // q 또는 Q를 입력하면 실행종료한다.
                   System.exit(0);
               } else if(command.equals("cd")) {
                   cd();
               } else {
                   for(int i=0; i < argArr.length;i++) {
                       System.out.println(argArr[i]);
                   }
               }
           } catch(Exception e) {
               e.printStackTrace();
               System.out.println("입력오류입니다.");
           }
       } // while(true)
   } // main

   public static void cd() {
```

```
        if(argArr.length==1) {
            System.out.println(curDir);
            return;
        } else if(argArr.length > 2) {
            System.out.println("USAGE : cd directory");
            return;
        }

        String subDir = argArr[1];

        /*
           (1) 아래의 로직에 맞게 코드를 작성하시오.
        1. 입력된 디렉토리(subDir)가 ".."이면,
            1.1 현재 디렉토리의 조상 디렉토리를 얻어서 현재 디렉토리로 지정한다.
                 (File클래스의 getParentFile()을 사용)
        2. 입력된 디렉토리(subDir)가 "."이면, 단순히 현재 디렉토리의 경로를 화면에 출력한다.
        3. 1 또는 2의 경우가 아니면,
            3.1 입력된 디렉토리(subDir)가 현재 디렉토리의 하위디렉토리인지 확인한다.
            3.2 확인결과가 true이면, 현재 디렉토리(curDir)을 입력된 디렉토리(subDir)로
                 변경한다.
            3.3 확인결과가 false이면, "유효하지 않은 디렉토리입니다."고 화면에 출력한다.
        */
    } // cd()
}
```

**[실행결과]**

```
C:\jdk1.8\work\ch15>java Exercise15_6
C:\jdk1.8\work\ch15>>
C:\jdk1.8\work\ch15>>cd ch15
유효하지 않은 디렉토리입니다.
C:\jdk1.8\work\ch15>>cd ..
C:\jdk1.8\work>>cd ch15
C:\jdk1.8\work\ch15>>
C:\jdk1.8\work\ch15>>cd .
C:\jdk1.8\work\ch15
C:\jdk1.8\work\ch15>>q

C:\jdk1.8\work\ch15>
```