

예제 8-7 : 버퍼 스트림을 이용한 출력

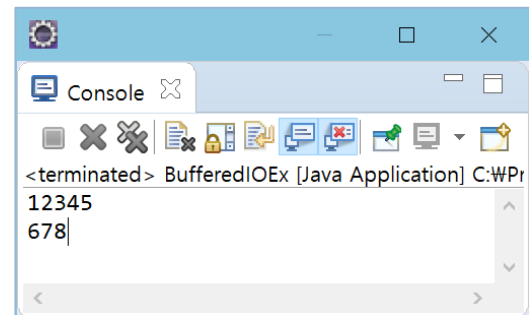
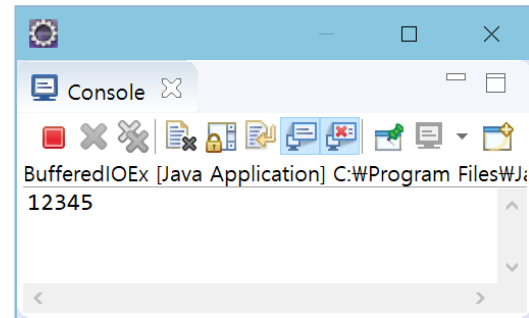
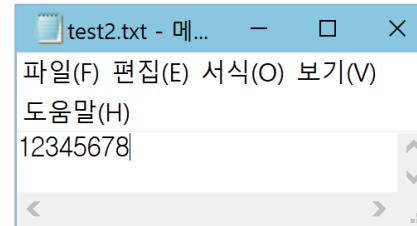
1

버퍼 크기를 5로 하고, 표준 출력 스트림(System.out)과 연결한 버퍼 출력 스트림을 생성하라. c:\WWTemp\test2.txt 파일을 저장된 영문 텍스트를 읽어 버퍼 출력 스트림을 통해 출력하라.

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

public class BufferedIOEx {
    public static void main(String[] args) {
        FileReader fin = null;
        int c;
        try {
            fin = new FileReader("c:\\WWTemp\\test2.txt");
            BufferedOutputStream out = new
                BufferedOutputStream(System.out, 5);
            while ((c = fin.read()) != -1) {
                out.write(c);
            }

            // 파일 데이터가 모두 출력된 상태
            new Scanner(System.in).nextLine();
            out.flush(); // 버퍼에 남아 있던 문자 모두 출력
            fin.close();
            out.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```



예제 8-8 : File 클래스 활용한 파일 관리

2

File 클래스를 이용하여 파일의 타입을 알아내고, 디렉터리에 있는 파일들을 나열하며, 디렉터리 이름을 변경하는 프로그램을 작성해보자.

```
import java.io.File;

public class FileEx {
    public static void listDirectory(File dir) {
        System.out.println("-----" + dir.getPath() +
            "의 서브 리스트 입니다.-----");
        File[] subFiles = dir.listFiles();
        for(int i=0; i<subFiles.length; i++) {
            File f = subFiles[i];
            long t = f.lastModified();
            System.out.print(f.getName());
            System.out.print("\t파일 크기: " + f.length());
            System.out.printf("\t수정한 시간: %tb %td %ta %tT\n",t,
                t, t, t);
        }
    }

    public static void main(String[] args) {
        File f1 = new File("c:\\windows\\system.ini");
        System.out.println(f1.getPath() + ", " + f1.getParent() + ", " +
            f1.getName());
        String res="";
        if(f1.isFile()) res = "파일";
        else if(f1.isDirectory()) res = "디렉토리";
        System.out.println(f1.getPath() + "은 " + res + "입니다.");
    }
}
```

```
File f2 = new File("c:\\\\Temp\\\\java_sample");
if(!f2.exists()) {
    f2.mkdir(); // 존재하지 않으면 디렉토리 생성
}
listDirectory(new File("c:\\\\Temp"));
f2.renameTo(new File("c:\\\\Temp\\\\javasample"));
listDirectory(new File("c:\\\\Temp"));
}
}
```

예제 8-9 : 텍스트 파일 복사

3

문자 스트림 `FileReader`와 `FileWriter`를 이용하여 `c:\windows\system.ini`를 `c:\Temp\system.txt` 파일로 복사하는 프로그램을 작성하라.

```
import java.io.*;

public class TextCopyEx {
    public static void main(String[] args){
        File src = new File("c:\\windows\\system.ini"); // 원본 파일 경로명
        File dest = new File("c:\\Temp\\system.txt"); // 복사 파일 경로명
        int c;
        try {
            FileReader fr = new FileReader(src);
            FileWriter fw = new FileWriter(dest);
            while((c = fr.read()) != -1) { // 문자 하나 읽고
                fw.write((char)c); // 문자 하나 쓰고
            }
            fr.close(); fw.close();
            System.out.println(src.getPath() + "를 " + dest.getPath() + "로 복사하였습니다.");
        } catch (IOException e) {
            System.out.println("파일 복사 오류");
        }
    }
}
```

예제 8-10 : 바이너리 파일 복사

4

바이트 스트림을 이용하여 바이너리 파일을 복사하는 프로그램을 작성하라

```
import java.io.*;

public class BinaryCopyEx {
    public static void main(String[] args) {
        File src = new File("c:\\Windows\\Web\\Wallpaper\\Theme1\\wimg1.jpg");
        File dest = new File("c:\\Temp\\copyimg.jpg");
        int c;
        try {
            FileInputStream fi = new FileInputStream(src);
            FileOutputStream fo = new FileOutputStream(dest);
            while((c = fi.read()) != -1) {
                fo.write((byte)c);
            }
            fi.close();
            fo.close();
            System.out.println(src.getPath() + "를 " +
                               dest.getPath() + "로 복사하였습니다.");
        } catch (IOException e) {
            System.out.println("파일 복사 오류");
        }
    }
}
```

예제 8-11 : 블록 단위로 바이너리 파일 고속 복사

5

예제 8-10을 10KB 단위로 읽고 쓰도록 수정하여 고속으로 파일을 복사하라.

```
import java.io.*;

public class BlockBinaryCopyEx {
    public static void main(String[] args) {
        File src = new File("c:\\Windows\\Web\\Wallpaper\\Theme1\\wimg1.jpg");
        File dest = new File("c:\\Temp\\desert.jpg");
        try {
            FileInputStream fi = new FileInputStream(src);
            FileOutputStream fo = new FileOutputStream(dest);
            byte [] buf = new byte [1024*10]; // 10KB 버퍼
            while(true) {
                int n = fi.read(buf); // 버퍼 크기만큼 읽기. n은 실제 읽은 바이트
                fo.write(buf, 0, n); // buf[0]부터 n 바이트 쓰기
                if(n < buf.length)
                    break;
            }
            fi.close();
            fo.close();
            System.out.println( src.getPath() + "를 " + dest.getPath() +
                               "로 복사하였습니다.");
        } catch (IOException e) { System.out.println("파일 복사 오류"); }
    }
}
```