

{ JAVA }

선린인터넷

웹 운영 과

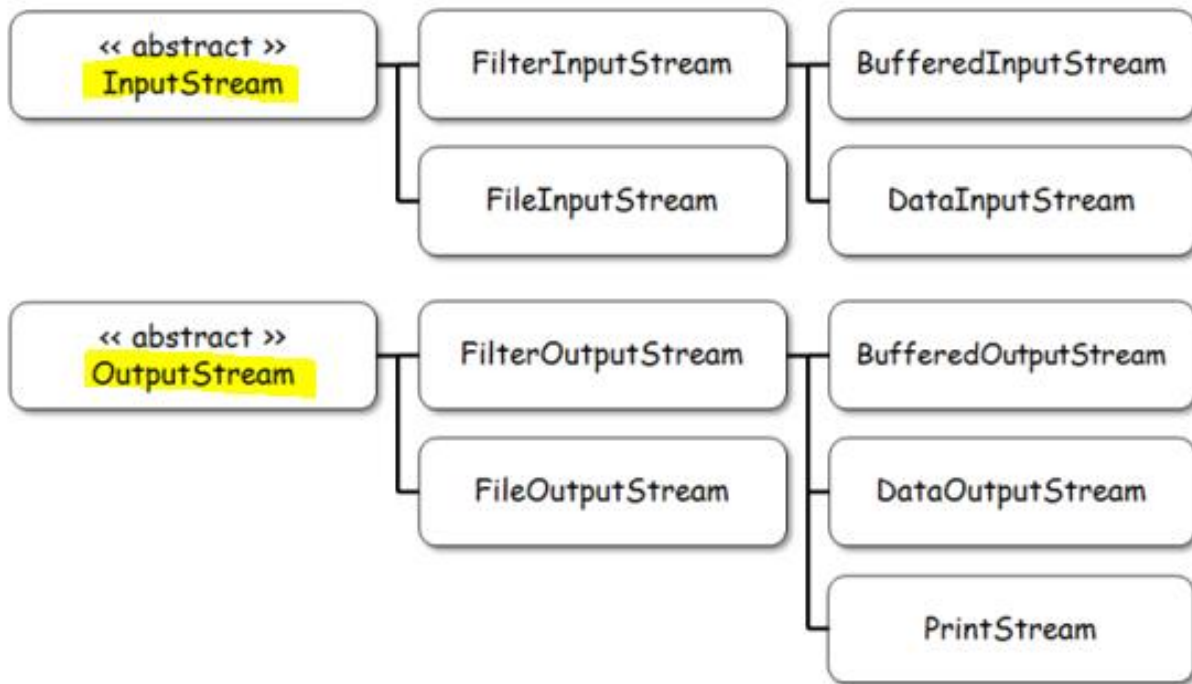
I/O 입출력

java.io 패키지 5

File 클래스

## 복습하기

1. 바이트(Byte) 스트림 : 1byte 단위로 읽고 쓰기 위한 스트림 : **read( ), write( )**



## 복습하기

## 2. StringTokenizer 클래스 : 문자열을 분리하기 위해 사용함

```
1 import java.util.StringTokenizer;
2
3 public class StExam3 {
4     public static void main(String[] args) {
5         StringTokenizer st =
6             new StringTokenizer("선린/인터넷/고등학교/웹운영과/2학년", "/");
7         while(st.hasMoreTokens())
8             System.out.println(st.nextToken());
9     }
10 }
```

Console

&lt;terminated&gt; StEx:

선린

인터넷

고등학교

웹운영과

2학년



## 복습하기

## 3. String 클래스 split( ) 메소드 : 문자열을 분리하기 위해 사용함

```
1 public class StExam4 {  
2     public static void main(String[] args) {  
3         String[] array  
4             = "선린/인터넷/고등학교/웹운영과/2학년".split("/");  
5         for (String s : array)  
6             System.out.println(s);  
7     }  
8 }
```

Console ✕  
<terminated> StEx:

선린  
인터넷  
고등학교  
웹운영과  
2학년



파일(디렉토리)에 대한 정보를 얻으려면...

## File 클래스

```
InputStream is = new FileInputStream
```

*FileInputStream*

🌟 `FileInputStream(File file)` - java.io.FileInputStream - 88%

🌟 `FileInputStream(String name)` - java.io.FileInputStream - 1

❖ File 클래스 : 생성자 및 주요 메소드

구분	생성자 또는 메소드	설명사항
생성자	File(String <b>fileName</b> ) File(String pathName , String <b>fileName</b> )	주어진 문자열(filename)을 이름으로 갖는 파일을 위한 File인스턴스(객체)를 생성함. 디렉토리도 같은 방법으로 다룸.
메소드	String getName( )	파일이름을 String으로 반환
	String getPath( )	파일 경로를 String으로 반환
	boolean exists( )	파일이 존재하면, true 반환
	boolean isFile( )	파일인지 확인하여 맞으면, true 반환
	boolean isDirectory( )	디렉토리인지 확인하여 맞으면, true 반환
	boolean createNewFile( )	아무런 내용이 없는 새로운 파일을 생성
	<b>String[ ]</b> list( )	현재 디렉토리의 파일/디렉토리 목록 반환
	<b>File[ ]</b> listFiles( )	현재 디렉토리의 파일/디렉토리 목록 반환

## ❖ File 클래스

- 운영체제마다 다른 파일 시스템에 대해 일관성 있게 파일에 접근하도록 만든 클래스
- 파일(디렉토리)에 대한 정보 및 처리를 담당한다.
- 내용 접근 : 불가능.
  - 즉, **파일 데이터를 읽고 쓰는 기능은 지원하지 않음** ⇨ **스트림 사용**
- File인스턴스(객체)를 생성했다고 해서 파일(디렉토리)가 생성되지 않음  
→ 존재하지 않는 **새 파일을 생성하려면...**  
  
File인스턴스 생성 후 **출력스트림 생성** 또는 **creatNewFile( ) 호출!**

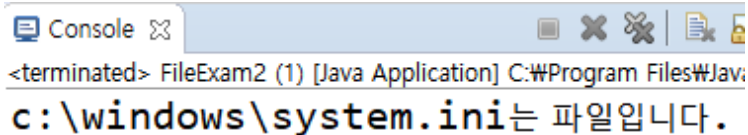
## 예제 1

```
1 import java.io.File;
2 public class FileExam1 {
3     public static void main(String[] args) throws Exception {
4         File file = new File("./src", "FileExam1.java"); // file 객체 생성
5         System.out.println("파일의 이름은? " + file.getName());
6         System.out.println("파일의 존재하는지? " + file.exists());
7         System.out.println("파일인가요? " + file.isFile());
8         System.out.println("디렉토리인가요? " + file.isDirectory());
9         System.out.println("파일의 바이트 크기는? " + file.length());
10
11         System.out.println("현재 디렉토리의 내용은? ");
12         String[] list = new File("./src").list();
13         for(int i=0; i<list.length; i++)
14             System.out.println(list[i]);
15     }
16 }
```



**예제2 : File/Directory 여부 구분하기, 경로 얻기 : 결과 출력 (system.ini에 대해서)****- File클래스 메소드**

**boolean `isFile( )`, boolean `isDirectory( )`, String `getPath( )`를 활용**


**- 출력예시**

```
<terminated> FileExam2 (1) [Java Application] C:\Program Files\Java\
c:\windows\system.ini는 파일입니다.
```

**- Hint** `import java.io.*;`

```
public class FileExam2 {
    public static void main(String[] args) {
        File f=new File("c:/windows/system.ini");
        String res = null;
        if( f. _____ ) res="파일";
        else if( f. _____ ) res="디렉토리";
        System.out.println( f. _____ + "는 " +res+"입니다.");
    }
}
```

**예제 3 :** java.zip 압축파일을 프로젝트 현재 폴더에 압축해제를 해봅시다.

▼  ilovejava

▼  src

▼  (default package)

>  FileExam1.java

>  FileExam2.java

>  FileExam3.java

>  FileExam4.java

>  JRE System Library [JavaSE-1.8]



java.zip

= Hint =

ZipInputStream 클래스 : getNextEntry( )

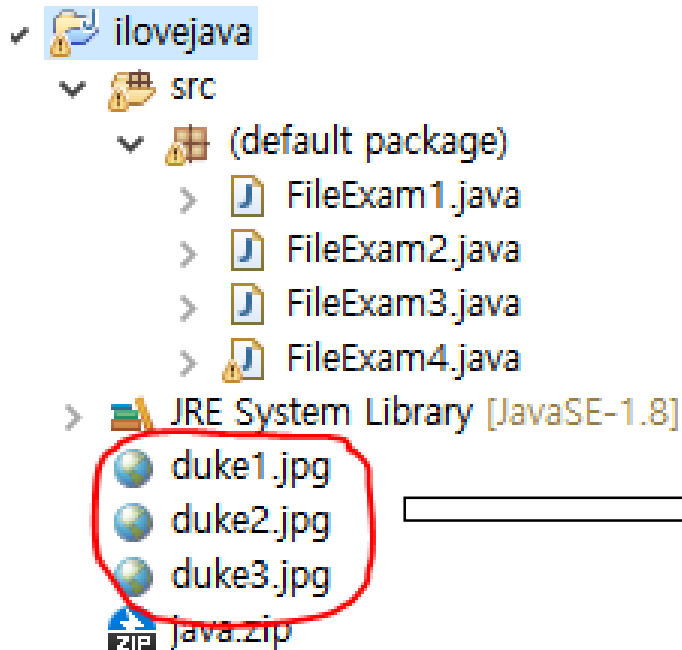
ZipEntry 클래스 : getName( )

java.zip를 풀어봅시다!

**예제 3 :** java.zip 압축파일을 프로젝트 현재 폴더에 압축해제를 해봅시다.

```
1 import java.io.*;
2 import java.util.zip.*;
3 public class FileExam3 {
4     public static void main(String[] args) throws Exception {
5         // 스트림 생성
6         File file = new File("java.zip");
7         ZipInputStream is = new ZipInputStream(new FileInputStream(file));
8         OutputStream os = null;
9         // 스트림 객체 메소드를 통한 복사
10        int input;
11        ZipEntry entry; // Zip파일 엔트리
12        while ((entry = is.getNextEntry()) != null) {
13            os = new FileOutputStream(entry.getName());
14            while ((input = is.read()) != -1)
15                os.write(input);
16        }
17        // 스트림 닫기
18        os.close();
19        is.close();
20    }
21 }
```

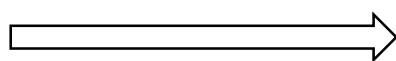
**예제 4 :** 여러 그림 파일들을 프로젝트 현재 폴더에 압축(result.zip)해봅시다.



= Hint =

ZipoutputStream 클래스 : putNextEntry( )

ZipEntry 클래스



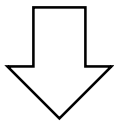
**result.zip로 압축해봅시다!**

**예제 4 :** 여러 그림 파일들을 프로젝트 현재 폴더에 압축(result.zip)해봅시다.

```
1 import java.io.*;
2 import java.util.zip.*;
3 public class FileExam4 {
4     public static void main(String[] args) throws Exception {
5         // 입력 스트림
6         String[] name = {"duke1.jpg", "duke2.jpg", "duke3.jpg"};
7         FileInputStream fi = null;
8         // zip 파일 출력 스트림
9         FileOutputStream fo = new FileOutputStream("result.zip");
10        ZipOutputStream zip = new ZipOutputStream(fo);
11
12        int input;
13        for(String n : name){
14            zip.putNextEntry(new ZipEntry(n));
15            fi=new FileInputStream(n);
16            while((input=fi.read())!=-1)
17                zip.write(input);
18        }
19        zip.close();
20        fi.close();
21    }
22 }
```

**예제 5 :** duke1.jpg 그림 파일을 흑백 이미지로 변환해서 gray.jpg로 저장해보자

= Hint = **픽셀 단위**로 흑백이미지로 변환



**BufferedImage** 클래스 : 픽셀 배열 이미지 처리  
getWidth( ), getHeight( ), getRGB( ), setRGB( )

**ImageIO** 클래스 : 이미지 파일을 읽고 쓰기  
read( ), write( )

**Color** 클래스 : RGB 타입의 색상을 나타냄  
getRed( ), getGreen( ), getBlue( )

**예제 5 : duke1.jpg 그림 파일을 흑백 이미지로 변환해서 gray.jpg로 저장해보자**

```
1 import java.awt.*;
2 import java.awt.image.BufferedImage;
3 import java.io.*;
4 import javax.imageio.ImageIO;
5 public class FileExam5 {
6     public static void main(String[] args) throws Exception {
7         File file = new File("duke1.jpg");
8         BufferedImage image = ImageIO.read(file);
9
10        for(int x=0; x<image.getWidth(); x++)
11            for(int y=0; y<image.getHeight(); y++){
12                Color col = new Color(image.getRGB(x, y));
13                int avg =
14                    (col.getRed() + col.getGreen() + col.getBlue()) / 3;
15                Color avgCol = new Color(avg, avg, avg);
16                image.setRGB(x, y, avgCol.getRGB());
17            }
18
19        File gray = new File("gray.jpg");
20        ImageIO.write(image, "jpg", gray);
21    }
22 }
```

**예제6** : 프로젝트 현재 폴더의 **파일 개수**, **디렉토리 개수**를 화면에 출력해보자

출력 예시 : 아래 숫자는 달라질 수 있습니다.

 Console 

<terminated> FileExam6 [J

파일 개수 : 8

디렉토리 개수 : 3



**예제6** : 프로젝트 현재 폴더의 **파일 개수**, **디렉토리 개수**를 화면에 출력해보자

```
1 import java.io.File;
2 public class FileExam6 {
3     public static void main(String[] args) {
4         int totalFile=0, totalDirectory=0;
5         File dir = new File("."); // 프로젝트 현재 폴더
6         // 해당 디렉토리의 파일목록을 읽어온다
7         File[] files = dir.listFiles();
8         // 파일 개수, 디렉토리 개수 카운트
9         for(File f: files){
10             if(f.isDirectory())
11                 totalDirectory++;
12             else
13                 totalFile++;
14         }
15         System.out.println("파일 개수 : "+ totalFile);
16         System.out.println("디렉토리 개수: "+ totalDirectory);
17     }
18 }
```

## 정리하기

### ❖ File 클래스

- 운영체제마다 다른 파일 시스템에 대해 일관성 있게 파일에 접근하도록 만든 클래스
- 파일(디렉토리)에 대한 정보 및 처리를 담당한다.
- 내용 접근 : 불가능.
  - 즉, 파일 데이터를 읽고 쓰는 기능은 지원하지 않음 ⇨ 스트림 사용

Char 스트림

