

{ JAVA }

선린인터넷

웹 운영 과

I/O 입출력

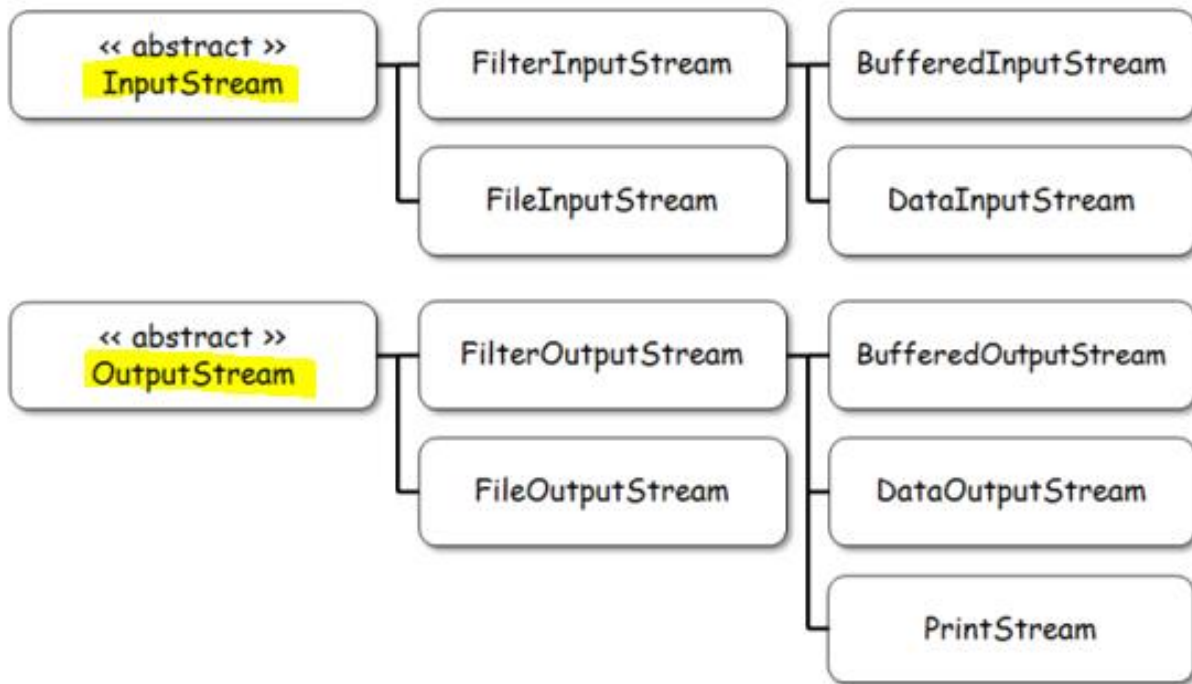
java.io 패키지 4

Byte Stream 실습

StringTokenizer

복습하기

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



실습문제 C:\windows\system.ini 파일의 내용을 모니터 화면에 출력하기

실행화면 예시

소스프로그램 파일명: Exam1.java

Console

<terminated> Exam1 (2) [Java Application] C:\Program File

```
; for 16-bit app support
[386Enh]
device=tddebug.386
woafont=dosapp.fon
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON

[drivers]
wave=mmdrv.dll
timer=timer.driv

[mci]
```

실습문제 C:\windows\system.ini 파일의 내용을 모니터 화면에 출력하기

실행화면 예시

```
Console
<terminated> Exam1 (2) [Java Application] C:\Program File
; for 16-bit app support
[386Enh]
device=tddebug.386
woafont=dosapp.fon
EGA80WOA.FON=EGA80WOA.FON
EGA40WOA.FON=EGA40WOA.FON
CGA80WOA.FON=CGA80WOA.FON
CGA40WOA.FON=CGA40WOA.FON

[drivers]
wave=mmdrv.dll
timer=timer.drv

[mci]
```

소스프로그램 파일명: Exam1.java

= Hint =

* 파일입력

FileInputStream 클래스의
read() 메소드를 활용해봅시다

* 화면출력

System.out.print() 메소드를
활용해봅시다.
단, 출력시 (char) 형변환 유의

실습문제 C:\Windows\system.ini 파일의 내용을 모니터 화면에 출력하기

read

```
public int read()  
    throws IOException
```

Reads the next byte of data from this input stream. The value byte is returned as an int in the range 0 to 255. If no byte is available because **the end of the stream has been reached, the value -1 is returned.** This method blocks until input data is available, the end of the stream is detected, or an exception is thrown.

This method simply performs `in.read()` and returns the result.

Specified by:

`read` in class `InputStream`

Returns:

the next byte of data, or -1 if the end of the stream is reached.

Throws:

`IOException` - if an I/O error occurs.

실습문제 C:\windows\system.ini 파일의 내용을 모니터 화면에 출력하기

소스프로그램 파일명: Exam1.java

```
1 import java.io.*;
2 public class Exam1 {
3     public static void main(String[] args) throws Exception{
4         // 스트림 생성
5         InputStream fi
6             = new FileInputStream("c:\\windows\\system.ini");
7         int inp;
8
9         // 스트림 메서드 호출하여 데이터 입출력하기
10        while( (inp=fi.read())!=-1 )
11            /* 모니터 화면에 출력하기 */
12
13        // 스트림 닫기
14        fi.close();
15    }
16 }
```

Console

<terminated> Exam1 (2) [Java Application] C:\Program File

; for 16-bit app support

[386Enh]

device=tddebug.386

woafont=dosapp.fon

EGA80WOA.FON=EGA80WOA.FON

EGA40WOA.FON=EGA40WOA.FON

CGA80WOA.FON=CGA80WOA.FON

CGA40WOA.FON=CGA40WOA.FON

[drivers]

wave=mmdrv.dll

timer=timer.drv

[mci]

교과서 118쪽 : 문자열을 분리하기 위해 사용하는 클래스 (**String** → 여러 **Token**들)

StringTokenizer 클래스

참고) String 클래스 : **split()** 메소드

❖ StringTokenizer 클래스

- **java.util** 패키지에 포함되어 있다.
- 문자열을 분리하기 위해 사용된다.
 - 구분문자(delimiter) : 문자열을 구분할 때 사용되는 문자
 - 토큰(token) : 구분 문자로 분리된 각각의 문자열 (나누는 단위가 되는 문자열)
 - String 클래스의 **split() 메소드**를 이용해도 동일한 기능을 구현 할 수 있다.

StringTokenizer stok = new StringTokenizer("사과 배 복숭아");

↑
토큰을 추출할 **문자열**을 가지고
StringTokenizer 객체를 **생성**한다

❖ StringTokenizer 클래스

구분	생성자 또는 메소드	설명사항
생성자	StringTokenizer(String s) StringTokenizer(Strings, String delim)	String을 파싱한 StringTokenizer 객체 생성 (delim : 구분문자 설정)
메소드	int countTokens()	StringTokenizer객체에 포함된 토큰의 개수
"	boolean hasMoreTokens()	StringTokenizer객체에 리턴할 다음 토큰이 있으면 true 값을 리턴
"	String nextToken()	파싱한 문자열을 차례대로 리턴 (반환)함

예제 1 : "사과 배 복숭아"를 공백기준으로 문자열을 분리해봅시다

```
1 import java.util.StringTokenizer;
2 public class StExam1 {
3     public static void main(String[] args) {
4         StringTokenizer st = new StringTokenizer("사과 배 복숭아");
5
6         String s1 = st.nextToken(); // 첫번째 토큰인 "사과" 리턴
7         System.out.println(s1);
8
9         String s2 = st.nextToken(); // 두번째 토큰인 "배" 리턴
10        System.out.println(s2);
11
12        String s3 = st.nextToken(); // 세번째 토큰인 "복숭아" 리턴
13        System.out.println(s3);
14    }
15 }
```

Console ✕
<terminated> StE

사과
배
복숭아

참고) 기본 구분문자: **공백문자**

- space 문자, tab 문자,
newline 문자,
carriage-return 문자, form-feed 문자

예제2 : "sunrin.hs.kr"를 마침표(".") 기준으로 문자열을 분리해봅시다

```
1 import java.util.StringTokenizer;
2 public class StExam2 {
3     public static void main(String[] args) {
4         StringTokenizer st = new StringTokenizer("sunrint.hs.kr", ".");
5
6         System.out.println("현재 토큰 개수: "+st.countTokens());
7         int i=0;
8         while(st.hasMoreTokens())
9             System.out.println(i++ + "번째: " + st.nextToken());
10
11         System.out.println("남은 토큰 개수: "+st.countTokens());
12     }
13 }
```

Console

<terminated> StExam1 (1) [Java A

현재 토큰 개수: 3

0번째: sunrint

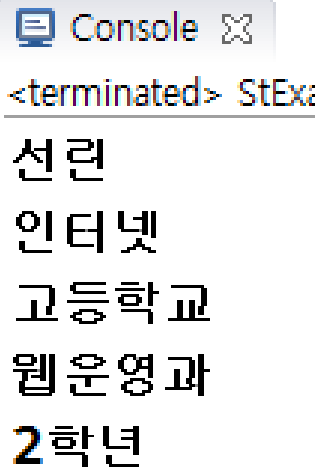
1번째: hs

2번째: kr

남은 토큰 개수: 0

예제 3 : “선린/인터넷/고등학교/웹운영과/2학년”을 “/”기준으로 문자열을 분리해서 모니터 화면에 차례대로 출력해봅시다.

소스프로그램 파일명: **StExam3.java**



Console ✖

<terminated> StExa

선린
인터넷
고등학교
웹운영과
2학년

= Hint =
StringTokenizer 클래스를
이용해봅시다

StringTokenizer

예제3 : “선린/인터넷/고등학교/웹운영과/2학년”을 “/”기준으로 문자열을 분리해서

Hint 모니터 화면에 차례대로 출력해봅시다. (아래 Hint 코드를 참고해보세요.)

소스프로그램 파일명: **StExam3.java**

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

Console

<terminated> StExam3

선린

인터넷

고등학교

웹운영과

2학년

예제 3 : “선린/인터넷/고등학교/웹운영과/2학년”을 “/”기준으로 문자열을 분리해서

예시 모니터 화면에 차례대로 출력해봅시다.

파일명: StExam3.java

```
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 }
```

StringTokenizer

예제4 : 예제3 프로그램을 String 클래스의 [split\(\) 메소드](#)를 이용하여 작성해보자.
선린/인터넷/고등학교/웹운영과/2학년"을 "/"기준으로 문자열을 분리해서 화면 출력
소스프로그램 파일명: **StExam4.java**

split

```
public String[] split(String regex)
```

Splits this string around matches of the given regular expression.

This method works as if by invoking the two-argument `split` method with the given expression and a limit argument of zero. Trailing empty strings are therefore not included in the resulting array.

The string "boo:and:foo", for example, yields the following results with these expressions:

Regex Result

```
: { "boo", "and", "foo" }  
o { "b", "", ":and:f" }
```

Parameters:

regex - the delimiting regular expression

Returns:

the array of strings computed by splitting this string around matches of the given regular expression

Console

<terminated> StExa

선린

인터넷

고등학교

웹운영과

2학년

예제4 : 예제3 프로그램을 String 클래스의 split() 메소드를 이용하여 작성해보자.
선린/인터넷/고등학교/웹운영과/2학년"을 "/"기준으로 문자열을 분리해서 화면 출력

예시

소스프로그램 파일명: **StExam4.java**

```
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> StExa

선린

인터넷

고등학교

웹운영과

2학년

예제5 : 문자열 "1+2+3+4+5+6+7+8+9+10"을 StringTokenizer 클래스와 Wrapper 클래스를 적절히 활용하여 그 **합**을 구하고 화면에 출력해보자.

소스프로그램 파일명: **StExam5.java**

Console

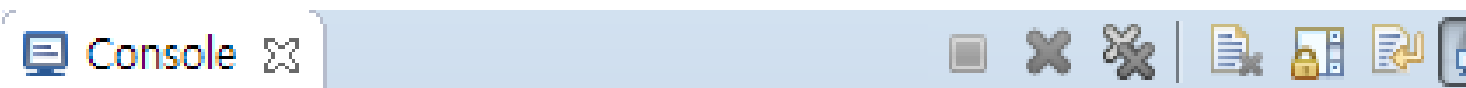
<terminated> StExam5 [Java Application] C:\Program Files\Java\jdk1.8.0_9

1+2+3+4+5+6+7+8+9+10의 합계는 **55**입니다.

예제5 : 문자열 "1+2+3+4+5+6+7+8+9+10"을 StringTokenizer 클래스와 Wrapper 클래스를 적절히 활용하여 그 합을 구하고 화면에 출력해보자.

소스프로그램 파일명: **StExam5.java**

= Hint =
Wrapper 클래스의
Integer.parseInt()메소드를
활용해봅시다



Console [Java Application] C:\Program Files\Java\jdk1.8.0_91

1+2+3+4+5+6+7+8+9+10의 합계는 55입니다.

예제5 : 문자열 "1+2+3+4+5+6+7+8+9+10"을 StringTokenizer 클래스와 Wrapper 클래스를 적절히 활용하여 그 **합**을 구하고 화면에 출력해보자.

소스프로그램 파일명: **StExam5.java**

```
1 import java.util.StringTokenizer;
2 public class StExam5 {
3     public static void main(String[] args) {
4         String str = "1+2+3+4+5+6+7+8+9+10";
5         // StringTokenizer 객체 생성 (구분자는 "+")
6
7         int sum = 0;
8         while( ) // StringTokenizer 객체에 토큰이 남아 있는지
9             sum = sum + Integer.parseInt( );
10
11         System.out.println(str + "의 합계는 " + sum + "입니다.");
12     }
13 }
```

= Hint =
Wrapper 클래스의
Integer.parseInt()
메소드를
활용해봅시다

File 클래스

