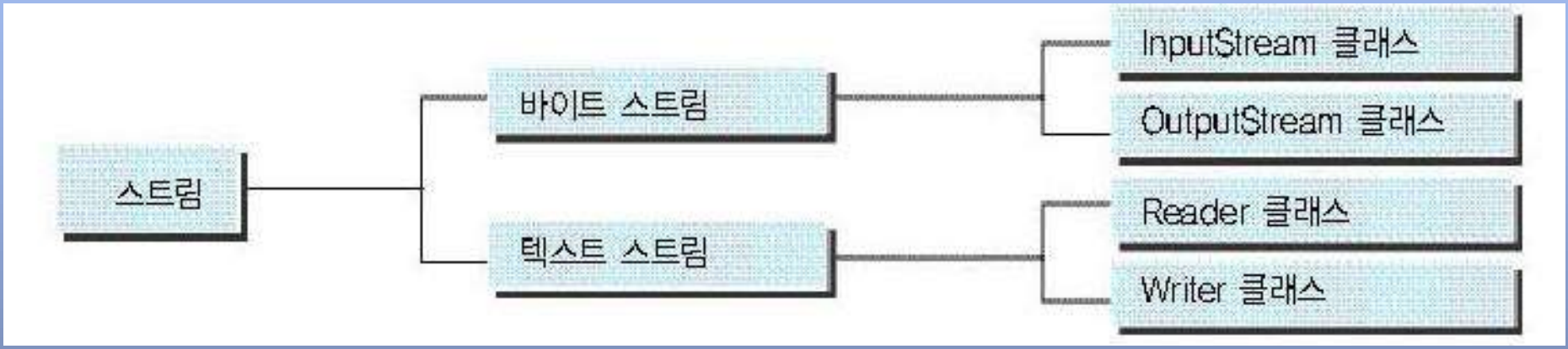
클릭하세요 자바

IO

- 자바에서의 입출력

최 문 환

1. 스트림(데이터를 운반하는데 사용되는 통로)



1. 스트림

바이트 스트림 클래스	특징	문자 스트림 클래스
InputStream	기본 입력 스트림 클래스	Reader
FileInputStream	파일 입력 스트림 클래스	FileReader
FilterInputStream	다른 필터 클래스의 최상위 클래스	FilteredReader
BufferedInputStream	버퍼 기능이 있어 편리	BufferedReader
DataInputStream	자바 기본형 데이터를 읽는데 편리	없음
OutputStream	기본 출력 스트림 클래스	Writer
FileOutputStream	파일 출력 스트림 클래스	FileWriter
BufferedOutputStream	버퍼 기능이 있어서 편리	BufferedWriter
DataOutputStream	자바 기본형 데이터를 출력할 때 유용	없음
PrintStream	표준 출력 시스템으로 나감	PrintWriter

2. InputStream 클래스

메서드	
int read()	한 바이트를 읽어 들인다.
int read(byte b[])	바이트 배열을 읽어 들인다.
int read(byte b[], int off, int len)	바이트 배열의 주어진 위치에 주어진 길이만큼 읽어 들인다.

<예제> 키보드로부터 한 글자를 입력받아 화면에 출력

```
001:public class IOTest00{
002:  public static void main(String[] args) throws Exception   {
003:    int data=0;          //키보드에서 입력받은 데이터를 저장할 변수
004:    System.out.println("문자를 입력하세요. 끝내려면 [Ctrl]+Z를 누르세요.");
005:    data = System.in.read();       //System.in   :   키보드와 연결되어 있는 통로
006:    while(data != -1){   //더 이상 읽을 값이 없으면([Ctrl]+Z를 누르면) -1를 리턴
007:      System.out.print((char)data);
008:      data = System.in.read();     //System.in   :   키보드와 연결되어 있는 통로
009:    }//while
010: }//main
011:}
```


<예제> 키보드로부터 한 글자를 입력받아 화면에 출력

```
001:import java.io.*;
002:public class IOTest01 {
003:  public static void main(String[] args) {
004:    int data=0;    //키보드에서 읽은 데이터를 저장할 변수 선언
005:
006:    System.out.println("문자를 입력하세요. 끝내려면 [Ctrl]+Z를 누르세요.");
007:    try{
008:      InputStream myIn=System.in; //키보드와 연결되어 있는 통로
009:      while( (data = myIn.read( )) != -1)   //더 이상 읽을 값이 없으면 -1를 리턴
010:        System.out.print((char)data); //입력받은 값을 출력
011:    }catch ( IOException   e ){
012:     e.printStackTrace();
013:   }//예외처리 끝
014:  }//main 끝
015:}
```


3. OutputStream 클래스

메서드	
void close()	스트림을 닫는다.
void flush()	출력 스트림이 갖고 있는 버퍼의 내용을 모두 출력 스트림으로 내보내고 비운다.
int write(byte[] b, int off, int len)	b 배열의 off 위치부터 len 만큼 출력한다.
void write(byte[] b)	바이트 배열로 출력한다.
void write(int b)	한 바이트로 출력한다.

3. OutputStream 클래스

<예제> 바이트출력 스트림 클래스를 사용한 예제

```
001:import java.io.*;
002:public class IOTest0A{
003:  public static void main(String[] args) throws Exception   {
004:    int data=0;           //키보드에서 입력받은 데이터를 저장할 변수 선언
005:    System.out.println("문자를 입력하세요. 끝내려면 [Ctrl]+Z를 누르세요.");
006:    OutputStream myOut = System.out;
007:    while((data = System.in.read()) != -1){
008:      myOut.write(data); //입력받은 값을 출력
009:    }//while
010: }//main
011:}
```

<예제> 입력된 문자열

```
001:import java.io.*;
002:public class IOTest02 {
003:  public static void main(String[] args) {
004:    int data=0; //키보드에서 읽은 데이터를 저장할 변수 선언
005:    int cnt=0;  //읽어 들인 문자열의 개수를 계산할 변수 선언
006:    InputStream myIn=System.in;        //표준 입력 장치인 키보드와 연결되어 있는 통로
007:    OutputStream myOut = System.out; //표준 출력 장치인 모니터와 연결되어 있는 통로
008:    System.out.println("문자를 입력하세요. 끝내려면 x혹은 X나 [Ctrl]+Z를 누르세요.");
009:    try{
010:      while((data = myIn.read()) != -1){   //더 이상 읽을 값이 없으면 -1를 리턴
011:        if (data=='x' || data=='X') //x나 X가 눌리면 종료
012:          break;
013:        cnt++; //입력 받은 문자의 개수를 센다.
014:        myOut.write(data); //입력받은 값을 출력
015:      }//while
016:    }catch(Exception e){
017:      e.printStackTrace();
018:    }//try
019:    System.out.println("₩n입력받은 총 갯수-> : " + cnt);
020:  }//main
021:}
```

4. File 클래스

생성자	
File(String directory) File(String directory, String file) File(File obj, String file)	directory : 파일 경로 file : 파일 이름 obj : File 객체
멤버 메서드	
boolean canRead()	파일이 읽기 가능하면 true, 아니면 false
boolean canWrite()	파일이 쓰기 가능하면 true, 아니면 false
boolean delete()	파일을 삭제하고 true 반환, 파일을 삭제할 수 없으면 false 반환
boolean exists()	파일이 존재하면 true, 아니면 false 반환
String getAbsolutePath()	파일 절대 경로 반환
String getCanonicalPath()	파일 정규 경로 반환
String getParent()	부모 디렉토리 이름 반환
String getPath()	파일 경로 반환

4. File 클래스

멤버 메서드	
String getName()	파일 이름 반환
boolean isDirectory()	디렉토리이면 true, 아니면 false 반환
boolean isFile()	파일이면 true, 아니면 false 반환
boolean isHidden()	파일 숨김 속성이면 true, 아니면 false 반환
boolean mkdir()	경로로 지정된 모든 부모 디렉토리가 존재해야함.
boolean mkdirs()	지정된 디렉토리가 존재하지 않으면 생성한 다음 지정한 디렉토리를 생성
boolean renameTo(File newname)	newname 으로 파일 이름을 변경. 성공하면 true, 실패하면 false
long lastModified()	1970년 1월 1일 0시부터 파일이 마지막으로 수정된 날짜까지의 시간을 밀리초 단위로 반환
long length()	파일 크기를 바이트로 반환
String [] list()	디렉토리에 있는 파일 목록 반환

<예제> 현재 디렉토리 파일 목록 출력하기

```
001:import java.io.File;
002:public class FileTest{
003:  public static void main (String args[]) {
004:    // 현재 디렉토리 출력
005:    File dir = new File (".");
006:    String[] strs = dir.list();
007:    for (int i = 0; i < strs.length; i++) {
008:      System.out.println (strs[i]);
009:    }
010:  }
011:}
```


5. FileInputStream 클래스

생성자	
FileInputStream(File file)	주어진 File 객체가 가리키는 파일을 바이트 스트림으로 읽기 위한 FileInputStream 객체를 생성
FileInputStream(String name)	주어진 이름이 가리키는 파일을 바이트 스트림으로 읽기 위한 FileInputStream 객체를 생성

5. FileInputStream 클래스

- 파일 생성하는 순서

```
FileInputStream fis=new FileInputStream("파일이름“);
data=fis.read( );
fin.close( );
```

```
File f=new File("파일이름“);
FileInputStream fis=new FileInputStream(f);
data=fis.read( );
fis.close( );
```

<예제>특정 파일의 내용을 화면에 출력

```
001:import java.io.*;
002:public class IOTest03{
003:  public static void main(String[] args) {
004:    int data=0;    //키보드에서 읽은 데이터를 저장할 변수 선언
005:    String path="c:/javawork/8bu/IOTest03.java";  //파일 경로와 이름 지정
006:
007:    try{
008:      FileInputStream fis=new FileInputStream(path);
009:      while( (data = fis.read( )) != -1)   //더 이상 읽을 값이 없으면 -1를 리턴
010:        System.out.write(data); //입력받은 값을 출력
011:    }catch (  IOException   e){
012:      e.printStackTrace();
013:   }//예외처리 끝
014:  }//main 끝
015:}
```

6. FileOutputStream 클래스

생성자	
FileOutputStream(File file)	주어진 File 객체가 가리키는 파일을 쓰기 위한 객체를 생성 기존의 파일이 존재할 때는 그 내용을 지우고 새로운 파일을 생성
FileOutputStream(String name)	주어진 이름의 파일을 쓰기 위한 객체를 생성
FileOutputStream(String name, boolean append)	주어진 append 값에 따라 새로운 파일을 생성하거나 또는 기존의 내용에 추가

6. FileOutputStream 클래스

▶ 파일 생성하는 순서

```
FileOutputStream fos = new FileOutputStream("파일이름“);
fos.write( 내용 );
fos.close();
```

▶ 파일 생성하는 순서

```
File f=new File("파일이름“);
FileOutputStream fos = new FileOutputStream(f);
fos.write( 내용 );
fos.close();
```

<예제> 키보드에서 입력받은 내용을 파일에 기록

```
001:import java.io.*;
002:class  FileOut{
003:  public static void main(String[] args)   {
004:    int data;
005:    System.out.println("저장할 내용을 입력하세요.  “);
006:    try{
007:      File f=new File("Test.txt");     //File 클래스의 인스턴스를 생성
008:      //FileOutputStream 은 무조건 해당 파일을 생성한다. 덮어쓴다.
009:      FileOutputStream fos= new FileOutputStream (f);
010:      while((data=System.in.read()) != -1) //키보드에서 데이터를 읽어 와서
011:        fos.write(data);                    //파일에 기록한다.
012:      fos.close(); //파일을 닫는다.
013:    }catch(FileNotFoundException fe)    {
014:      System.out.println(fe.getMessage());
015:    }
016:    catch(IOException ie)    {
017:      ie.printStackTrace();
018:    }
019:  }
020:}
```