

18장 입출력과 네트워크

입출력

자바프로그램내부 데이터 받아온다 = 입력

키보드 파일 메모리버퍼

자바프로그램외부로 데이터를 내보낸다 = 출력

모니터 파일 메모리버퍼

스트림 = stream = 데이터(1BYTE / 2BYTE(char - 단일문자(2byte) = unicode A-65 a -97) 연속 흐름

java.io.*

	입력스트림	출력스트림
1바이트(바이트스트림) - 영문자, a.jpg-이미지 a.mp3-음향	abstract class InputStream{read() ; 구현 (){} } XXXXInputStream class FileInputStream extends InputStream { read() {구현} } read() - 데이터 1바이트씩 입력 close()-finally블록 처리	OutputStream xxxOutputStream write(1바이트데이터) close()
2바이트(문자스트림) - 다국어	Reader xxReader read()- 데이터 2바이트씩 입력 close()	Writer xxxWriter write(2바이트데이터) write(String- 1개문자 2바이트) close()
File - win 탐색기, mac 파일관리자 --> 파일이나 디렉토리 취급 기능 자바에서 구현		

- 콘솔 입출력

콘솔 입력 - 키보드 입력 - 표준입력

콘솔 출력 - 모니터 출력- 표준출력 -->

System.out--> 현재 자바 실행 시스템 모니터 변수

System.out.print()->현재 자바 실행 시스템 모니터로 출력

System.in --> 현재 자바 실행 시스템 키보드 변수

System.in.read() --> 현재 자바 실행 시스템 키보드 1byte 입력 기능 메소드 호출

==> 자바 데이터타입 사용 불가능 . 한글 변환 불가능.

DataInputStream 이용하면 데이터타입 변환 사용 가능

InputStreamReader 이용하면 한글

```
class System{
    static InputStream in;
    static PrintStream out; > OutputStream 상속
}
```

- java.util.Scanner 클래스 -> jdk 동일 기능이 여러 클래스 제공
java.util --> "사용 편리 도와"

```
Scanner s = new Scanner(System.in);
int result = s.nextInt();
double s = s.nextDouble();
boolean b = s.nextBoolean();
nextxxxx()
String result1 =s.next();// 한개 단어-문자열( 문자열 내부 모든 문자들은 2바이트)
String s = s.nextLine();// 1줄 입력 .String
```

System.in -> 콘솔 입력
Scanner s = new Scanner(System.in);

18.4 파일 입출력

FileInputStream	FileOutputStream
FileReader	FileWriter
File	

- File 클래스

용도 – os 파일시스템(=파일과 디렉토리) 관리 기능 이용 제공. 입출력 기능 메소드 없다
생성자

File f = new File("c:/test/a.txt"); -> 모든os이클립스 c:\test\Wa.txt 파일 존재 간주

File f = new File("c:\test\Wa.txt"); -> window이클립스 c:\test\Wa.txt 파일 존재 간주

File f = new File("./a.txt"); -> 이클립스 현재javaproject 루트디렉토리 a.txt 파일 존재 간주

File f = new File("a.txt"); -> 이클립스 현재javaproject 루트디렉토리 a.txt 파일 존재 간주

File f = new File("../a.txt"); -> 이클립스 workspace디렉토리 a.txt 파일 존재 간주

File f = new File("c:/java"); -> 디렉토리

File f = new File("."); -> 현재디렉토리

File f = new File(".."); -> 상위디렉토리

파일이나 디렉토리 구분자 - '/'

파일이나 디렉토리 2가지

c:/..... --> 절대경로

../a.java --> 상대경로

- 메소드

isFile() / isDirectory() --> boolean

canRead() --> 읽기가능 ->boolean

canWrite() --> 쓰기가능->boolean

length() --> 파일 총 크기(byte단위)-->int

getXXXXPath();-->파일 모든 경로-->String

lastModified() --> 파일최종수정시각정보--> 1/1000초단위 -->long

324324325454-->Calendar / Date / SimpleDateFormat

String s[] = list(); --> 디렉토리내부 세부 내용(파일과 디렉토리)

exists() 파일 존재 여부

읽기전용

읽기쓰기

-FileInputStream

용도 - 파일 내용 입력. 1바이트

생성자

FileInputStream fi = new FileInputStream("입력파일명");

FileInputStream fi = new FileInputStream("/*현재디렉토리생략*/입력파일명");

FileInputStream fi = new FileInputStream("c:/java/입력파일명");

FileInputStream fi = new FileInputStream("../..../입력파일명");

메소드

read()

fi.read(); -> 파일 첫바이트 입력

int result = fi.read(); 유니코드 2바이트 / ascii code 1바이트(영문자들 숫자 특수문자몇개)

result ==> -1 (end of file-eof

while(true){

int result = fi.read();

if(result == -1){ break; }

}

fi.close();

<pre> FileInputStream fi = null; try{ fi = new FileInputStream("입력파일명"); while(true){ int result = fi.read(); if(result == -1){ break; } } }catch(IOException e){ e.printStackTrace... }finally{ fi.close(); } </pre>	<pre> FileOutputStream fi = null; try{ fi = new FileOutputStream("출력파일명", true); 3문자 byte= -128 ~ 127 정수만 fi.write(65) ; 'A' fi.write(97) ; 'a' fi.write(13); \n }catch(IOException e){ e.printStackTrace... }finally{ fi.close(); } </pre>
<pre> FileReader fi = null; try{ fi = new FileReader("입력파일명"); while(true){ int result = fi.read(); ->2바이트씩 한글 if(result == -1){ break; } } }catch(IOException e){ e.printStackTrace... }finally{ fi.close(); } </pre>	<pre> FileWriter fi = null; try{ fi = new FileWriter("출력파일명", true); 3문자 byte= -128 ~ 127 정수만 fi.write(65) ; 'A' fi.write(97) ; 'a' fi.write(13); \n fi.write('가'); fi.write("가나다"); }catch(IOException e){ e.printStackTrace... }finally{ fi.close(); } </pre>

a.txt 파일을 b.txt 파일 복사

반복 - a.txt 파일 입력 – 결과를 b.txt 출력

'/' -> slash --> linux solaris mac-파일 분리자

'\w' -> windows - \w\w-파일분리자

자바- 모든 os 동작한다

c:\w\w\w\w

메소드

```
data.txt
{
  100 200
}
```

1> data.txt 파일 입력

FileInputStream / FileReader

read(); 7번 1 0 0 공백 2 0 0

readInt()-->존재x

2>100 + 200 ==> 300

파일/키보드
파일/모니터

Scanner
데이터타입 가공

자바
source stream
보조스트림

DataInputStream / Out

BufferedIn / Out / Reader / Writer

Object

==> 자바 데이터 타입 변환 / 한글 변환

Scanner sc = new Scanner(System.in);

Scanner sc = new Scanner(new FileInputStream("")));

Scanner sc = new Scanner(new FileReader("")));

sc.nextInt()

java.net.*

