

-리뷰

18장 java.io패키지의 클래스들

표준 장치=물리적장치/하드웨어=키보드입력,모니터출력 / 파일 등의 입출력 기능 / 파일시스템 정보 추출

System.in->java.io.InputStream타입 System.in.read() ->키보드 입력 - 1byte(영문 숫자 킵 드특수문자.. - 한글, 다국어 표현 불가	System.out -> java.io.PrintStream타입 System.out.print() / println()
java.util.Scanner Scanner s = new Scanner(System.in); s.nextInt() s.nextDouble() s.next() / s.nextLine()	

- 파일

java.io.FileReader(한글) java.io.FileInputStream (한글 깨져서 / 한글 제대로) windows-파일출력-a.txt a.txt-다른컴퓨터- 파일입력	java.io.FileWriter java.io.FileOutputStream	java.io.File - 파일 시스템 정보 제공 - 입출력 기능 메소드 없다 - windows 탐색기 기능
입력 int r = read() close()	출력 FileWriter write(char c[]) / write(String) FileOutputStream write(byte b[]) close()	

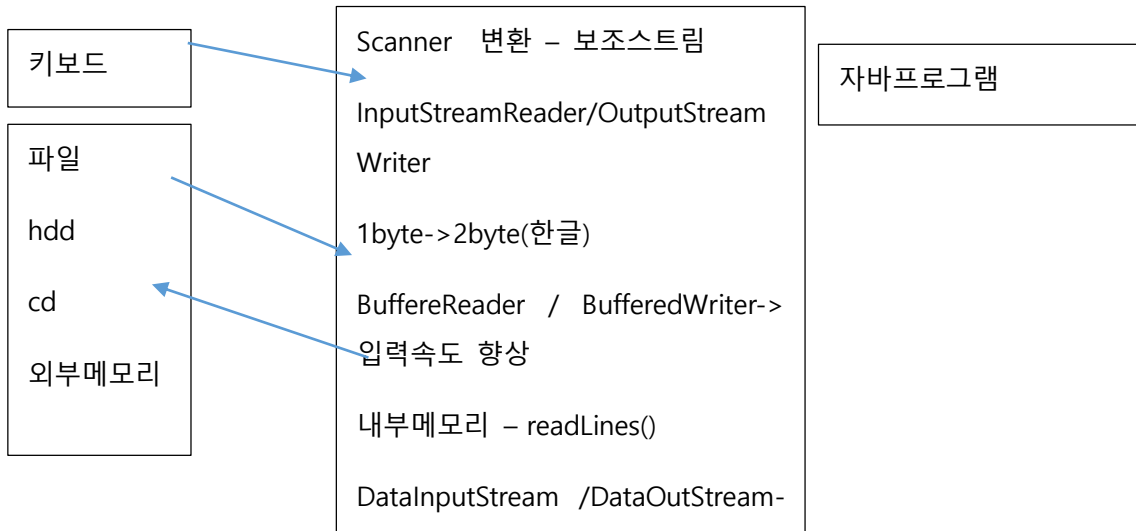
- 18.5 보조스트림

1028~1052

System.in.read()

Scanner s = new Scanner(System.in);

입력데이터출발지=원본



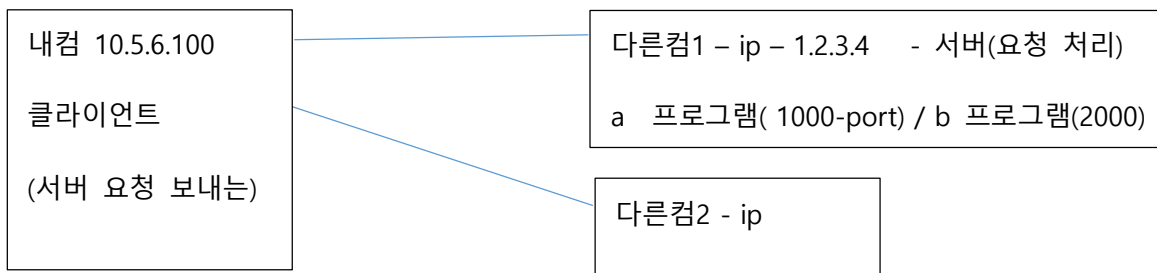
java.io.50여개 클래스 -> 복잡 -> 여러개 클래스 기능 1-2개 클래스 사용 -> jdk 1.5 Scanner

10시 40분

18 장 네트워크 기능

java.net 패키지

네트워크 - 물리적 통신 회선으로 연결된 다른 컴퓨터와 입출력통신



1.ip address-연결 컴퓨터 식별번호 / 인터넷상 모든 컴퓨터 고유한 식별 번호 (중복x)

IPv4 주소: 10.5.6.100

(0-255).(0-255).(0-255).(0-255)

256*256*256*256 최대갯수

ipv6

2.port – 1개 컴퓨터 동작 프로그램 식별번호

ip + port

0-65535($2^{16}=65536$ 개)

java.net.InetAddress 클래스 – ip

- 통신

tcp 통신 방식 - 전화방식	udp 통신 방식 - 우편방식
java.net.ServerSocket java.net.Socket 나 – 건다 – 내용 – 끊는다 -친구1	java.net.DatagramSocket java.net.DatagramPacket 나 – (내용, 누가, 누구한테)-친구1

- 이미 구현 제공 네트워크 방식

인터넷 = 웹 = http / 파일 전송 특화 = ftp / 원격 접속 = telnet

http://www.xxx.xxx

protocol = 통신 컴퓨터 약속 규칙

전화

로그인서버역할프로그램	로그인클라이언트프로그램
<p>1. 서버 시작한다.(port) 0-65535 <code>ServerSocket ss = new ServerSocket(9999);</code></p> <p>3. 클라이언트 요청 승인한다. <code>Socket s = ss.accept();</code></p> <p>5. 클라이언트가 전송한 아이디와 암호를 입력받아서 처리한다(가입, 암호 맞는지...) <code>byte [] id_byte_server = new byte[100];</code> <code>InputStream is = s.getInputStream();</code> <code>is.read(id_byte_server);</code> <code>--> {'m'숫자코드, 'u', 'l' 't', 'i'}</code></p> <p><code>String id = new String(id_byte_server); //"muli"</code></p> <p>6. 처리결과(정상, 아이디 x, 암호 x....)를 클라이언트에 게 전송한다. <code>OutputStream os = s.getOutputStream();</code> <code>os.write(.....);</code></p> <p>9. 클라이언트와 접속 해제한다 <code>s.close();</code></p>	<p>2. 서버 접속을 요청한다.(서버 ip, 서버 port) <code>Socket s = new Socket(ip, 9999);</code></p> <p>4. 서버로 로그인아이디와 암호 전송한다. <code>String id = "multi";</code> <code>byte [] id_byte = id.getBytes(); -> {'m'숫자코드, 'u', 'l' 't', 'i'}</code> <code>OutputStream os = s.getOutputStream();</code> <code>os.write(id_byte);</code></p> <p>7. 서버가 전송한 결과를 이용한다. <code>InputStream is = s.getInputStream();</code> <code>???<--is.read();</code></p> <p>8. 서버와의 접속 해제한다. <code>s.close();</code></p>
<code>InputStream is = s.getInputStream();</code> <code>is.read(byte[]);</code> 바이트배열 -> String	String-->바이트배열 변환 후 <code>OutputStream os = s.getOutputStream();</code> <code>os.write(byte[]);</code>

서버- 무한

클라이언트1

클라이언트2

클라이언트3

[java.net.ConnectException](#): Connection refused: connect -> 서버 아직 시작되지 않았는데 클라이언트가 접속 상태 (해결책 서버 먼저 시작)

[java.net.BindException](#) : ADDRESS IS AL....이미 사용중 -> 서버 이미 시작한 상태에서 서버 또 실행 상태(포트 번호 중복) (해결책 이미 실행중 1개 서버만 사용)

클라이언트

LoginClient.java

키보드

아이디: multi

암호: 1111

서버

LoginServer.java

클라이언트가 키보드 입력하여 전송한 아이디와 암호를 받아서 적합성을 검사한다

```
HashMap<String, String> users =
```

```
new HashMap<String, String>();
```

```
users.put("multi", "campus"); //key -아이디 value-암호
```

```
users.put("java", "program");
```

```
users.put("oracle", "db");
```

users 해당 아이디가 없다 - "회원가입부터 하세요" 클라이언트 출력

users 해당 아이디가 있고 암호는 다르다 - "암호를 확인하세요" 클라이언트로 출력

users 해당 아이디가 있고 암호는 맞다 - "정상 로그인 사용자입니다" 클라이언트로 출력