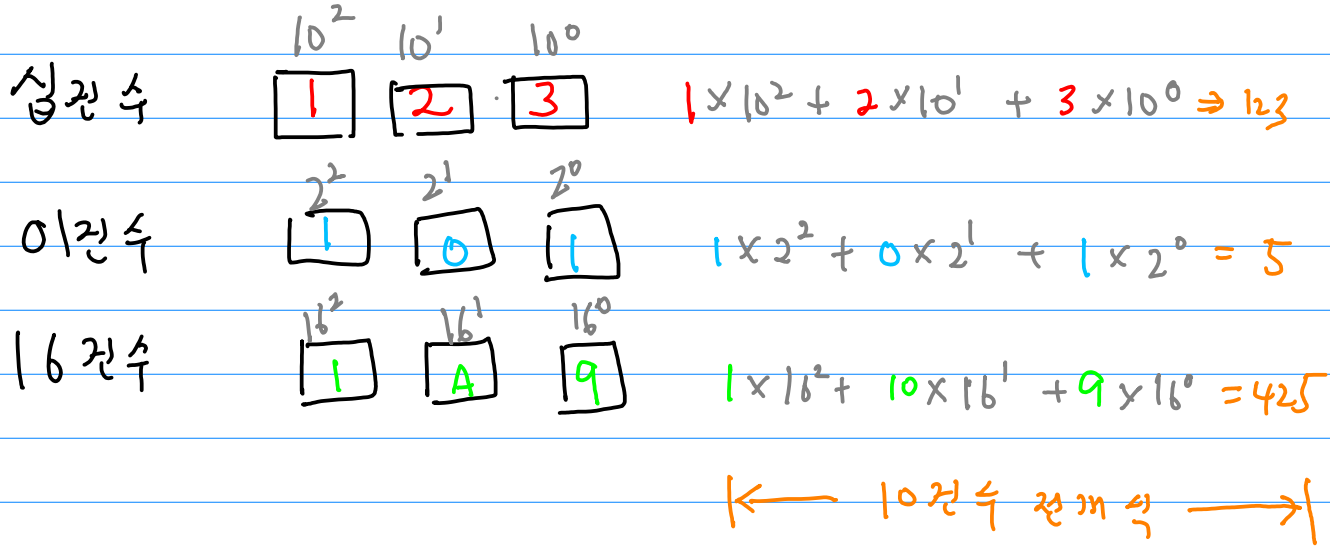
Day13 (H1)

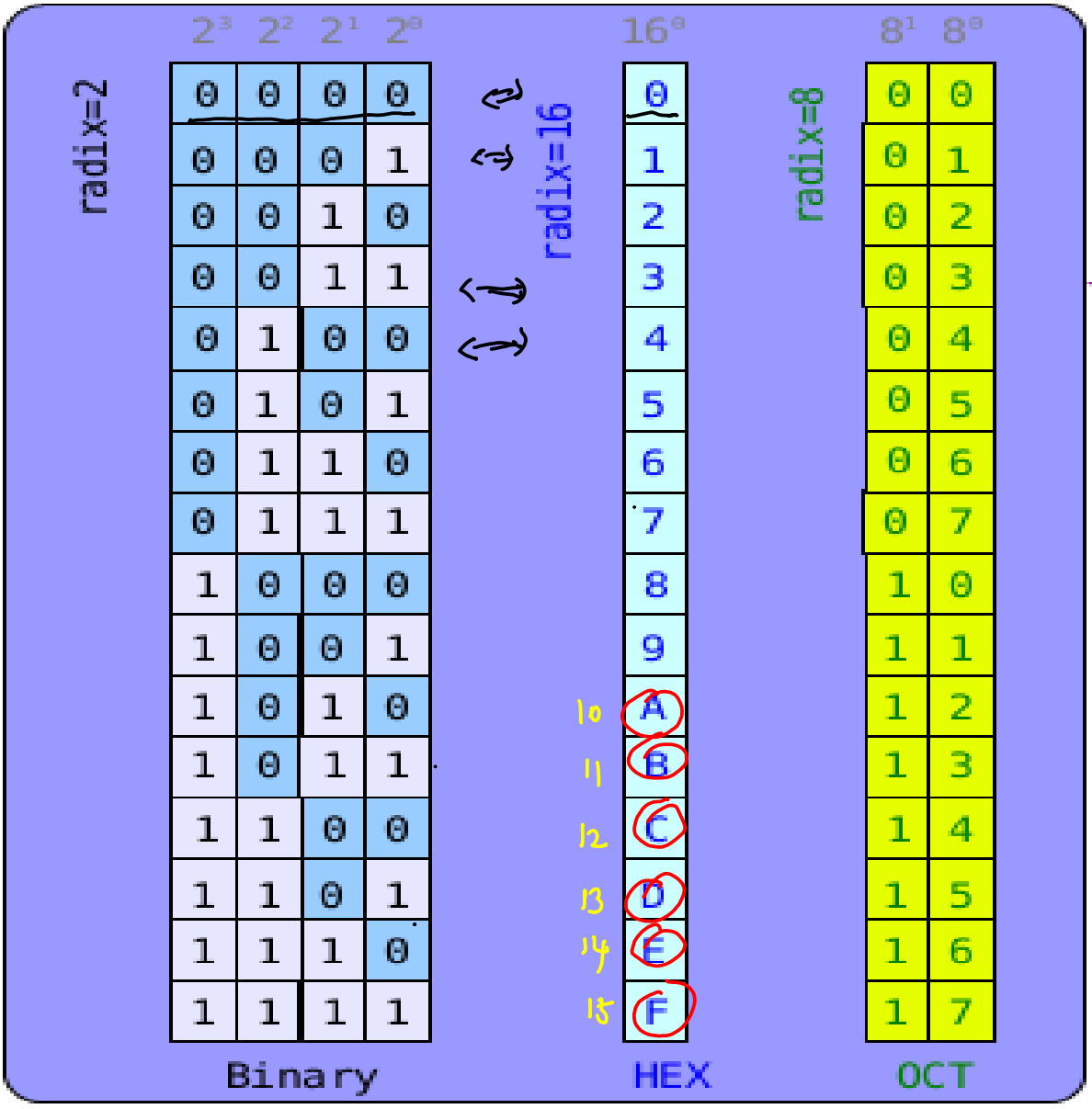
File IO

20150827

Binary, Hexa Decimal, Decimal



8 4 2 1


radix=2
2^3 2^2 2^1 2^0
0 0 0 0
0 0 0 1
0 0 1 0
0 0 1 1
0 1 0 0
0 1 0 1
0 1 1 0
0 1 1 1
1 0 0 0
1 0 0 1
1 0 1 0
1 0 1 1
1 1 0 0
1 1 0 1
1 1 1 0
1 1 1 1
Binary
radix=16
16^0
0
1
2
3
4
5
6
7
8
9
10 A
11 B
12 C
13 D
14 E
15 F
HEX
radix=8
8^1 8^0
0 0
0 1
0 2
0 3
0 4
0 5
0 6
0 7
1 0
1 1
1 2
1 3
1 4
1 5
1 6
1 7
OCT

Java에서 16진수 표기

4비 (11)

숫자앞에 0X 접두사

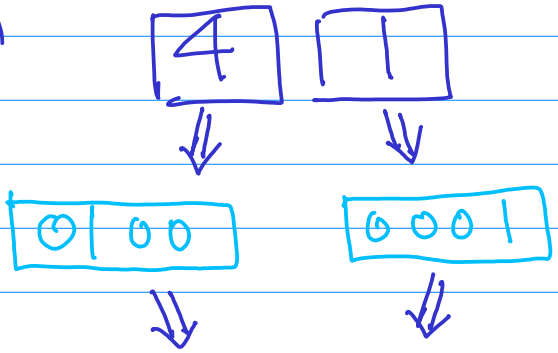
0X 41 ⇒ 4×16^1 1×16^0 (16)

16진수

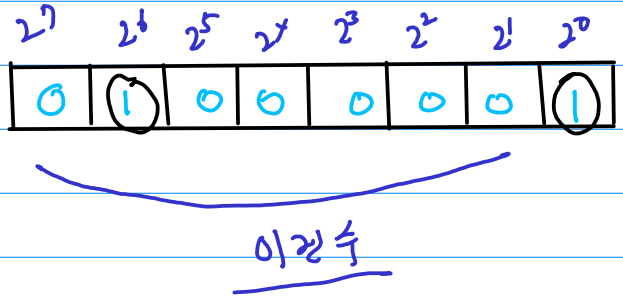
$4 \times 16^1 + 1 \times 16^0 = 65$

십진수

16진수



16진수 한 자씩 → 이진수로 변환

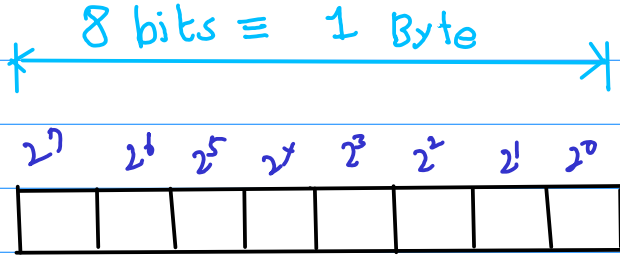


이진수

$1 \times 2^6 + 2^0 = 64 + 1 = 65$

십진수

11 de



① 숫자 $(2^0 \sim 2^8-1)$ 양수
$(-2^7 \sim +2^7-1)$ 정수 ⊕⊖

② 문자 ASCII 코드로 정해져있음
0x41 ⇔ A
0x42 ⇔ B

- Byte Stream : machine formatted data
  읽을수 있는 형태가 아님
  - InputStream
  - OutputStream
- Character Stream : human readable data
  읽을수 있는 형태임
  - Reader
  - Writer

File Read & Write

Byte Stream

- FileInputStream → InputStream
- FileOutputStream → OutputStream

Char Stream

- FileReader → Reader
- FileWriter → Writer

Class 이름들 / 조상 class

java.io package

```
import java.io.*;
```

Exception

Exception … 예외 상황 error …

FileNotFoundException

파일 존재하지 않을 때 Exception

Exception 을 던지고 (throws)

받아서 처리 (catch)

Byte Stream

```java
import java.io.*;

public class FileTest {

    /**
     * @param args
     */
    public static void main(String[] args) throws IOException {
        // TODO Auto-generated method stub

        FileOutputStream out = new FileOutputStream("out.txt");

        out.write('H');
        out.write('e');
        out.write('l');
        out.write('l');
        out.write('o');

        out.close();

    }

}
```

① import java.io.*

② Exception handle
Exception throws

Byte -128 ~ +127 정수

1-byte 크기의 정수

Byte Stream

~~0XFF~~

FileOutputStream
FileInputStream

- write()
- read()

Character Stream

FileWriter
FileReader

- write()
- read()

out2 charstream

out Bytestream

```
young@young-Samsung-NB-System:~/workspace/Day13$ ls
bin  in2.txt  out2.txt  out.txt  src
young@young-Samsung-NB-System:~/workspace/Day13$ ls -l
total 20
drwxrwxr-x 2 young young 4096  8월 26 15:15 bin
-rw-rw-r-- 1 young young   48  8월 26 16:18 in2.txt
-rw-rw-r-- 1 young young   10  8월 26 21:33 out2.txt
-rw-rw-r-- 1 young young    6  8월 26 21:33 out.txt
drwxrwxr-x 2 young young 4096  8월 26 15:15 src
young@young-Samsung-NB-System:~/workspace/Day13$
```

참조 변수

byte stream file

실제 파일 이름

① 객체 생성

```
FileOutputStream out  = new FileOutputStream("out.txt");
FileWriter       out2 = new FileWriter      ("out2.txt");
```

char stream file

```
out.write(0x41);  // ASCII code : A
out.write(0x42);  // ASCII code : B
out.write(0x43);  // ASCII code : C
out.write(0x44);  // ASCII code : D
out.write(0x45);  // ASCII code : E

byte b[] = {0x46, 0x47, 0x48, 0x49};
out.write(b);
```

② write()

```
out2.write('a');
out2.write('b');
out2.write('c');
out2.write('d');
out2.write('e');

out2.write("fghij");
```

out2 →

```
::::::::::::::
out2.txt
::::::::::::::
abcdefghij
::::::::::::::
out.txt
::::::::::::::
ABCDEFGHI
```

out →

```
out.close();
out2.close();
```

② close()

length = 4

byte b[] = {0x46, 0x47, 0x48, 0x49};

배열 초기값

byte[] b = new byte[4];

```
b[0] = 0x46;
b[1] = 0x47;
b[2] = 0x48;
b[3] = 0x49;
```

Scanner class를 사용하기 위해서

import java.util.*

Char stream

```
FileReader in2 = new FileReader("in2.txt");

Scanner    s   = new Scanner( in2 );
Scanner    s2  = new Scanner( System.in );


int x;

while (s.hasNext()) {
    x = s.nextInt();
    System.out.println( x );
}

x = s2.nextInt();
System.out.println(x);
```

파일에 있는 item이 남아 있으면 true

integer 읽기

in2

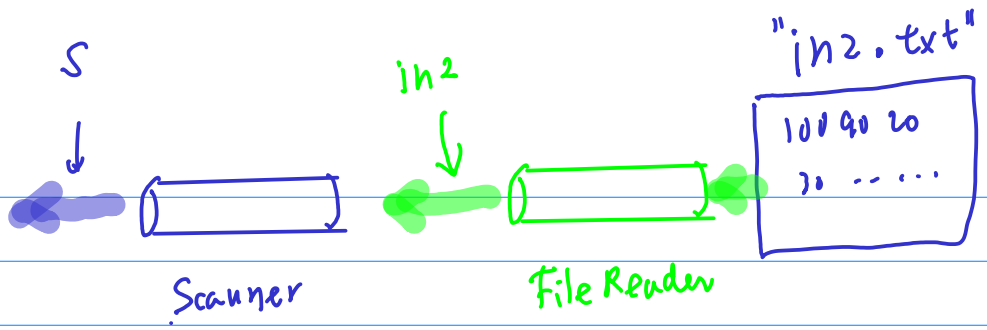
```
100 82 72
88 95 56
77 92 55
100 100 90
80 70 90
```

"in2.txt" file

```
100
82
72
88
95
56
77
92
55
100
100
90
80
70
90

123
123
```

key board input




```
FileReader in2 = new FileReader("in2.txt");

Scanner    s   = new Scanner( in2 );
Scanner    s2  = new Scanner( System.in );


int x;

while (s.hasNext()) {
    x = s.nextInt();
    System.out.println( x );
}

x = s2.nextInt();
System.out.println(x);
```