Lec6
2025-07-15 5:15 PM

data Type and Type Casting

Number System

① decimal → 10
② Binary → 2
③ Octal → 8
④ Hexa → 16

10 → 0-9
2 → 0,1
8 → 0-7
16 → 0-9, A-F

$9438 = 9\times10^3 + 4\times10^2 + 3\times10^1 + 8\times10^0$

9/5=1 (5×1=0)

$\frac{9}{5}=1$
$\frac{4}{5}=1$
$\frac{3}{5}=1$

1111
+ 2343 → 5 bits
2343
1023 0 → 5 bits

5 2 8
2 3 4 7
8 1 . 5

15 → 0.5
$\frac{15}{30}=\frac{5}{10}$
$\frac{1}{2}=0.5$

1/2=0.5
$\frac{1}{2}=0.5$
$\frac{1}{2}=0.5$
$\frac{1}{2}=0.5$

Conversion

while(n>0)
int rem = n%2
n=n/2
3

2	49	1
2	24	0
2	12	0
2	6	0
2	3	1
1	1	1
	0	

100101
11001

1000

rem=1

while(n>0)
int rem = n%2
sum = sum + rem*10
mul = mul * 10
n=n/2
3

2	49	1 × 10^0
2	24	0 × 10^1
2	12	0 × 10^2
2	6	0 × 10^3
2	3	1 × 10^4
1	1	1 × 10^5
	0	

$1\times10^0 + 0\times10^1 + 0\times10^2 + 0\times10^3 + 1\times10^4 + 1\times10^5$

rem = 38456

while(n>0)
int rem = n%2
sum = sum + rem*10
mul = mul * 10
n=n/2
3

Sum = $1\times10^0 + 1\times10^1 + 1\times10^2 + 0\times10^3 + 1\times10^4$

2	23	1 × 10^0
2	11	1 × 10^1
2	5	1 × 10^2
2	2	0 × 10^3
2	1	1 × 10^4
	0	

100111

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n = 39;
    int sum = 0;
    int rem = n%2; // 39%2
    while (n > 0) {
        int rem = n % 2;
        sum = sum + rem * mul;
        mul = mul * 10;
        n = n / 2;
    }
    System.out.println(sum);
}
```

Going to bed

2	23	1 × 10^0
2	11	1 × 10^1
2	5	1 × 10^2
2	2	0 × 10^3
2	1	1 × 10^4
	0	

100111

$Sum = 1\times10^0 + 1\times10^1 + 1\times10^2 + 0\times10^3 + 1\times10^4$

10000
110
10
1
10111

while(n>0)
rem = n%10
sum = sum*10 + rem
mul = mul*2
n=n/10
3

10	101010	
10	10101	0
10	1011	1
10	101	1
10	10	1
10	1	0
	0	1

$\frac{1}{2}$ → 1
0 × 10
$\frac{2}{2}$ → 2
$\frac{1}{2}$ → 0
0 × 10^2
$\frac{2}{2}$ → 1
$\frac{1}{2}$ → 0
1 × 10^3
$\frac{2}{2}$ → 1
$\frac{1}{2}$ → 0
1 × 10^4

55

```
public static void main(String[] args) {
    // TODO Auto-generated method stub
    int n=38511;
    int mul = 1;
    int sum=0;
    while (n > 0) {
        int rem = n % Des_Base;
        sum = sum + rem * mul;
        mul = mul * src_Base;
        n = n / Des_Base;
    }
    System.out.println(sum);
}
```

2	67	1 × 10^0
2	33	1 × 10^1
2	16	0 × 10^2
2	8	0 × 10^3
2	4	0 × 10^4
2	2	0 × 10^5
2	1	1 × 10^6
	0	

1000011

Binary → Octal
Octal → Binary
Binary → Hexa
Hexa → Binary
Octal → Hexa
Hexa → Octal

10 → 2

$\frac{7}{2}=3$
$\frac{3}{2}=1$

+ 1111
101010
101011
= 1011000

2/3=0
2/2=1
3/2=1
1/2=0
1/2=0

$\frac{5}{2}=2$

data Types

Primitive | Non-Primitive

Primitive:
int → 4 bytes → -2^31 to 2^31-1
short → 2 bytes → -32,768 to 32,767
long → 8 bytes → -2^63 to 2^63-1
float → 4 bytes → 7 digits
double → 8 bytes → 15 digits
char → 2 bytes
boolean → 1 bit

Non-Primitive:
byte → 1 byte → -128 to 127
int → 4 byte → -2^31 to 2^31-1
long → 8 byte → -2^63 to 2^63-1

int x=5;
byte y=5;

1 1 1 1 1 1 1 1 → 255

00000000 → 0
11111111 → 255

Signed int → 255

10 → Signed byte

1 1 1 1 1 1 1 1 → -1
1 1 1 1 1 1 1 0 → -2
1 1 1 1 1 1 0 1 → -3
1 1 1 1 1 0 0 0 → -8
1 1 1 1 1 1 1 1 → -128

-128 to 127

1 0 0 0 0 0 0 0 → -128

10 → Signed byte

1 0 0 0 0 0 0 0 → -128
0 0 0 0 0 0 0 0 → 0
0 0 0 0 0 0 0 1 → 1
0 0 0 0 0 0 1 0 → 2
0 0 0 0 0 1 0 0 → 4
0 0 0 0 1 0 0 0 → 8
0 0 0 0 1 1 1 1 → 15
0 0 0 0 0 0 0 0 → 0
1 1 1 1 1 1 1 1 → -1

1 0 0 0 0 0 0 0 → -128
1 1 1 1 1 1 1 1 → -1
1 0 0 0 0 0 0 0 → -128

10 → 1 byte → 8 bit → -128 to 127 → $\left[-2^7 \text{ to } 2^7-1\right]$

short → 16 bit → -2^{15} to $2^{15}-1$

int → 32 bit → -2^{31} to $2^{31}-1$

long → 64 bit → -2^{63} to $2^{63}-1$

Short C.T
int x=5

byte b=5

int b=-5

00000 101 → System में 2's Complement होगा

00000101
11111011
11111011 → System में -5 होगा 2's Complement होगा

byte b = (byte) (420);

420 → 1 1 0 1 0 1 1 0 0

$2^8 + 2^7 + 2^6 + 2^4 + 2^2$

256

MSB → 1 1 0 1 0 1 1 0 0
0 1 0 1 0 1 1 1
0 1 0 1 0 1 0 0 → -84

byte b = (byte) (300);

300 → 1 0 0 1 0 1 1 0 0

0010 1100 → 44

MSB → 0 → 44

$2^8 + 2^5 + 2^4 + 2^2$

256

256 + 16 + 4 = 52 + 10 = 62

'a' → 65
'b' → 66
'c' → 67
'd' → 68

'a' → 97
'b' → 98
'c' → 99
'd' → 100

2 . 2

Class → Started at 5:15 PM