?

2017. 04. 05.

김상영

Ubiquitous Computing Technology Research Institute
SungKyunKwan University

변수

- 기본형(primitive type) 변수

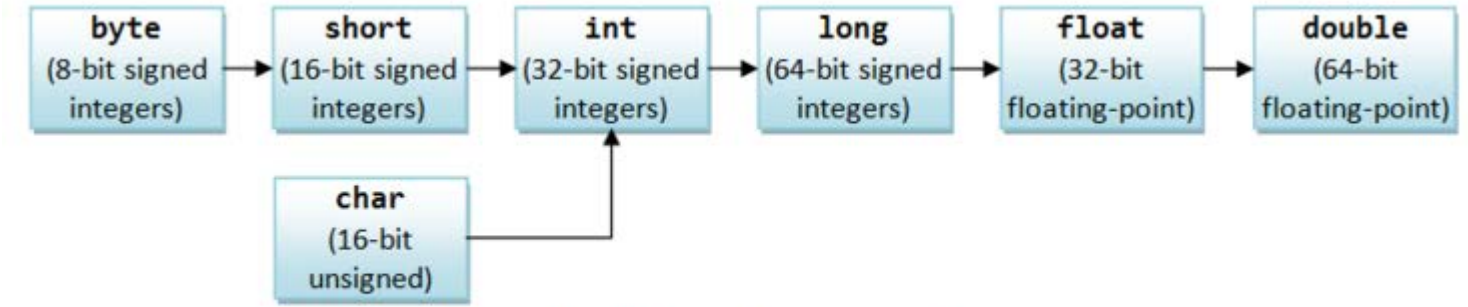
Name	Data	Range	Default Value	Size
byte	signed integer	[-128, 127]	0	8 bits
short	signed integer	[-32768, 32767]	0	16 bits
int	signed integer	[-2147483648, 2147483647]	0	32 bits
long	signed integer	[-9223372036854775808, 9223372036854775807]	0	64 bits
float	floating-point	**MIN:** ±1.4E-45 **MAX:** ±3.4028235E+38	0.0	32 bits
double	floating-point	**MIN:** ±4.9E-324 **MAX:** ±1.7976931348623157E+308	0.0	64 bits
char	Unicode	['\u0000', '\uFFFF']	'\u0000'	16 bits
boolean	logical value	{false, true}	false	≥ 1 bit

- 참조형(Reference type) 변수

  - 8개의 기본형을 제외한 나머지 타입, 객체의 주소를 저장.

형변환(casting)

- 공식



Orders of Implicit Type-Casting for Primitives

- 기본형 – 기본형 (단, Object class나 wrapper class는 가능)
- 참조형 – 참조형
- Example

```
int score = (int) 85.4;
byte b = (byte) score;
```

연산자(Operator)

- 연산자(Operator) 어떠한 기능을 수행하는 기호(+,0,*,/ 등)
- 피연산자(Operand) 연산자의 작업 대상(변수, 상수….)
- 우선순위
  - 산술 > 비교 > 논리 > 대입

종 류	연산방향	연산자	우선순위
단항 연산자	←	++ -- + - ~ ! (타입)	높음
산술 연산자	→	* / %	
	→	+ -	
	→	<< >> >>>	
비교 연산자	→	< > <= >= instanceof	
	→	== !=	
논리 연산자	→	&	
	→	^	
	→	|	
	→	&&	
	→	||	
삼항 연산자	→	?:	
대입 연산자	←	= *= /= %= += -= <<= >>= >>>= &= ^= |=	낮음

연산자의 종류와 우선순위

단항 연산자

- 증감연산자 (++, --)

증가연산자(++) – 피연산자 값을 1 증가

감소연산자(--) – 피연산자 값을 1 감소

전위형 – j=++i;

후위형 – j=i++;

```
1.  /* package whatever; // don't place package name! */
2.
3.  import java.util.*;
4.  import java.lang.*;
5.  import java.io.*;
6.
7.  /* Name of the class has to be "Main" only if the class is public. */
8.  class Ideone
9.  {
10.     public static void main (String[] args) throws java.lang.Exception
11.     {
12.         // your code goes here
13.         int i = 3;
14.         int j = 5;
15.         System.out.println(i++);
16.         System.out.println(++j);
17.         System.out.println("i=" + i + ", j=" + j);
18.     }
19. }
```

- ‘++i’ vs ‘i=i+1’

- 결과는 서로 같지만 실제로는 연산이 수행되는 과정은 다르다.
- 컴파일된 코드가 ++i가 훨씬 더 적기 때문에 더 빠르게 수행.

논리부정 연산자(!)

```
 8  class Ideone
 9  {
10      public static void main (String[] args) throws java.lang.Exception
11      {
12          // your code goes here
13          boolean _operator = false;
14          System.out.println(_operator);
15          _operator = !_operator;
16          System.out.println(_operator);
17          _operator = !_operator;
18          System.out.println(_operator);
19      }
```

stdout

```
false
true
false
```

비교연산자

```java
/* Name of the class has to be "Main" only if the class is public. */
class Ideone
{
    public static void main (String[] args) throws java.lang.Exception
    {
        // your code goes here
        if(10 == 10.0f) System.out.println("true");
        if('0'!= 0) System.out.println("true");
        if('A'==65) System.out.println("true");

        int num=5;
        if(num > 0 && num < 10) System.out.println("true");

        float f= 0.1f;
        double d = 0.1;
        double d2 = (double)f;

        System.out.println(f);
        System.out.println(d);
        System.out.println(d2);
        System.out.println((d==f));
        System.out.println((d==d2));
        System.out.println((d2==f));
    }
}
```

stdout

```
true
true
true
true
0.1
0.1
0.10000000149011612
false
false
true
```

Call by value? Call by reference?

```
8.  class Ideone
9.  {
10.     private static void swap(int a, int b) {
11.         int temp = a;
12.         a = b;
13.         b = temp;
14.     }
15.
16.     public static void main (String[] args) throws java.lang.Exception
17.     {
18.         // your code goes here
19.         int a = 1;
20.         int b = 2;
21.
22.         System.out.println("a => " + a);
23.         System.out.println("b => " + b);
24.
25.         swap(a, b);
26.
27.         System.out.println("------- swap -------");
28.
29.         System.out.println("a => " + a);
30.         System.out.println("b => " + b);
31.
32.     }
33. }
```

stdout

```
a => 1
b => 2
------- swap -------
a => 1
b => 2
```

Call by value? Call by reference?

```
8.  class Ideone
9.  {
10.     private static void swap(Integer a, Integer b) {
11.         Integer temp = a;
12.         a = b;
13.         b = temp;
14.     }
15.
16.     public static void main (String[] args) throws java.lang.Exception
17.     {
18.         // your code goes here
19.         Integer a = new Integer(1);
20.         Integer b = new Integer(2);
21.
22.         System.out.println("a => " + a.intValue());
23.         System.out.println("b => " + b.intValue());
24.
25.         swap(a, b);
26.
27.         System.out.println("------- swap -------");
28.
29.         System.out.println("a => " + a.intValue());
30.         System.out.println("b => " + b.intValue());
31.
32.     }
33. }
```

stdout

```
a => 1
b => 2
------- swap -------
a => 1
b => 2
```

Call by value? Call by reference?

```
8.  class Ideone
9.  {
10.     int value;
11.     Ideone(int value){
12.         this.value = value;
13.     }
14.     private static void swap(Ideone a, Ideone b) {
15.         int temp = a.value;
16.         a.value = b.value;
17.         b.value = temp;
18.     }
19.     public static void main (String[] args) throws java.lang.Exception
20.     {
21.         // your code goes here
22.         Ideone a = new Ideone(1);
23.         Ideone b = new Ideone(2);
24.         System.out.println("a => " + a.value);
25.         System.out.println("b => " + b.value);
26.
27.         swap(a, b);
28.
29.         System.out.println("------- swap -------");
30.         System.out.println("a => " + a.value);
31.         System.out.println("b => " + b.value);
32.     }
33. }
```

stdout

```
a => 1
b => 2
------- swap -------
a => 2
b => 1
```

실습문제

- 1번
  - 0~100 사이의 난수(num1), 70~90사이의 난수(num2), num1부터 num2 사이의 숫자 중에서 3의 배수이거나 7의 배수인 숫자들의 합을 구하여 출력하자.
  - 조건: 함수사용, 전역변수 사용금지.
  - 출력형식: 합 = sum

- 2번
  - 0~100 사이의 난수(num1), 70~90사이의 난수(num2), num1부터 num2 사이의 숫자 중에서 8의 배수이거나 12의 배수인 숫자들의 합을 구하여 출력하자
  - 조건: 함수사용, 전역변수 사용금지.
  - 출력형식: 합 = sum