3

chapter3_정리

1. 연산자

- 연산자(Operator)
 - 연산자는 연산을 수행하는 기호를말한다.
- 연산자와 피연산자
 - 연산자가 연산을 수행하려면 반드시 연산의 대상이 있어야 하는데, 이것을 피연산 자라고 한다.
 - 피연산자로는 상수, 변수 또는 식 등을 사용할 수 있다.
- 연산자의 종류

종류	연산자	설명	
산술 연산자	+ - * / % << >>	사칙 연산(+,-,*,/)과 나머지 연산(%)	
비교 연산자	> < >= <= == !=	크고 작음과 같고 다름을 비교 '그리고(AND)'와 '또는(OR)'으로 조건을 연결	
논리 연산자	&& ! & ^ ~		
대입 연산자	=	우변의 값을 좌변에 저장	
기타	(type) ?: instanceof	형변환 연산자, 삼항 연산자, instanceof연산자	

▲ 표3-1 연산자의 기능별 분류

• 산술 변환

이항 연산자는 두 피연산자의 타입이 일치해야 연산이 가능하므로, 피연산자의 타입이 서로 다르다면 형변환을 통해 일치 시켜야 한다.

ex)int 타입과 float타입을 덧셈하는 경우

2. 단항 연산자

- 증감 연산자 ++
 - ++ 피연산자의 값을 1 증가
 - 피연산자의 값을 1 감소

전위형 - 값이 참조되기 전에 증가, 감소

후위형 - 값이 참조된 후에 증가, 감소

- 부호 연산자 + -
 - boolean형과 char형을 제외한 기본형에만 사용 가능하다.

3. 산술 연산자

- 사칙연산자와 나머지 연산자가 있다.
 - 사칙 연산자 + * /
 - 피 연산자가 정수형의 경우 나누는 수를 0으로 사용하면 에러가 발생한다.
 - 0.0f, 0.0d로 나누는 것은 가능하지만 결과값은 무한대이다.
 - 형변화

```
byte a = 30;
```

byte b = 10;

byte c = (byte)(a*b);

sout (c);

결과 값은 44이다.

a * b 에서 결과 값이 300이지만 형변환을 통해 값 손실이 일어난다.

- int 형변환
- char 형변화
- 나머지 연산자

4. 비교 연산자

- 비교 연산자는 두 피연산자를 비교하는데 사용되는 연산자이다.
 - 대소비교 연산자

<> ≤≥

• 등가비교 연산자

== ≠

• 문자열 비교 - equals()를 쓴다.

5. 논리 연산자

- 논리 연산자 &&, ||, !
- 비트 연산자

&, |, ^, ~, <<, >>

- | (OR연산자) - 피연산자 중 한 쪽의 값이 1이면, 1을겨로가로 얻는다. 그 외에는

2

chapter3_정리

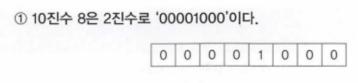
0

- & (AND연산자) 피연산자 양 쪽이 모두 1 이어야만 1을결과로 얻는다. 그외에는 0
- ^ (XOR연산자) 피연산자의 값이 서로 다를 때만 1을 결과로 얻는다. 같을 때는 0

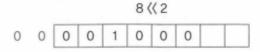
х	у	xly	х&у	x ^ y
1	1	1	1	0
1	0	1	0	1
0	1	1	0	1
0	0	0	0	0

▲ 표3-14 비트 연산자의 연산결과

- ~ 전환 연산자
 피연산자를 2진수로 표현했을 때, 0은 1로, 1은 0으로 바꾼다.
- 쉬프트 연산자
 - 피연산자의 각 자리를 오른쪽으로 또는 왼쪽으로 이동한다 해서 쉬프트 연산자라고 한다.



② '8 〈〈 2'은 10진수 8의 2진수를 왼쪽으로 2자리 이동시킨다.



③ 자리이동으로 인해 저장범위를 벗어난 값은 버려지고, 빈자리는 0으로 채워진다.



④ '8 〈〈 2'의 결과는 2진수로 '00100000'이 된다.(10진수로 32)



x << n 은 x * 2의 n제곱의 결과와 같다. x >> n 은 x / 2의 n제곱의 결과와 같다.

• 조건 연산자 ? - 삼항 연산자

chapter3_정리 4