



Java : Operator



Operator

연산자

데이터를 가공하기 위해 사용하는 기호 집합
기존의 데이터를 연산자를 통해 계산하여 원하는 값을 얻기 위해 사용

연산자 우선 순위

일반적으로 프로그램은 왼쪽에서 오른쪽, 위에서 아래로 코드가 진행된다
연산자가 사용되면 **실행 순서가 달라질 수 있다**



Operator

- 연산자 우선 순위

최우선 연산자	.(점) , [], ()
단항 연산자	!, ~, +/-, ++/--, (cast 자료형), instanceof
산술 연산자	+, -, *, /, %
Shift 연산자	<<, >>, >>>
관계 연산자	<, >, <=, >=, ==, !=
비트 연산자	&, , ^
논리 연산자	&&,
삼항 연산자	(조건항) ? 참 항 : 거짓 항
배정 대입 연산자	=, *=, /=, %=, +=, -=, <<=, >>=, >>>=
증감 후위 연산자	++/--
순차 연산자	,(콤마)



Operator

산술 연산자 : 데이터 간의 산술연산에 사용되는 연산자

산술 연산자	사용 예	의 미
+	$a + b$	두수의 합
-	$a - b$	두수의 차
*	$a * b$	두수의 곱
/	a / b	나누기 몫
%	$a \% b$	나누기 나머지



Operator

관계 연산자 : 데이터 간의 크기 비교 및 값 일치 여부에 사용
(비교 관계 연산자, 항등 관계 연산자로 구분)

관계 연산자	사용 예	결과 (논리형)
<code>a > b</code>	<code>10 > 20</code>	false
<code>a >= b</code>	<code>10 >= 20</code>	false
<code>a < b</code>	<code>10 < 20</code>	true
<code>a <= b</code>	<code>10 <= 20</code>	true
<code>a == b</code>	<code>10 == 20</code>	false
<code>a != b</code>	<code>10 != 20</code>	true



Operator

단항 연산자 : 값의 앞 혹은 뒤에서 위치함

(부호 구분 및 증감 연산자로 구분, 증감 연산자의 증가 및 감소값은 1로 고정)

단항 연산자	사용 예	결과 (논리형)
+ , -	<code>int b = -a</code>	a : 10, b : -10
++ (전치)	<code>++num</code>	선 증가, 후 처리
-- (전치)	<code>--num</code>	선 감소, 후 처리
++ (후치)	<code>num++</code>	선 처리, 후 증가
-- (후치)	<code>num--</code>	선 처리, 후 감소



Operator

배정 대입 연산자 : 데이터를 유지한 채로 산술 연산을 수행한다

대입 연산자	사용 예	의 미
<code>+=</code>	<code>a += b</code>	<code>a = a + b</code>
<code>-=</code>	<code>a -= b</code>	<code>a = a - b</code>
<code>*=</code>	<code>a *= b</code>	<code>a = a * b</code>
<code>/=</code>	<code>a /= b</code>	<code>a = a / b</code>
<code>%=</code>	<code>a %= b</code>	<code>a = a % b</code>



Operator

논리 연산자 : 양쪽 데이터의 논리 결과에 따라 true / false 값을 나타낸다

```
boolean bo1 = true;  
boolean bo2 = false;
```

논리 연산자	사용 예	결과 (논리형)
&& (AND 연산)	bo1 && bo2	false
	bo1 && true	true
 (OR 연산)	bo1 bo2	true
	bo2 false	false



Operator

삼항 연산자 : 조건식에 따라 참인 경우 앞부분, 거짓이면 뒷부분을 수행

```
int n1 = 10;
int n2 = 20;

int n3 = (n1 > n2) ? n1 : n2;
// (조건식) ? 참 : 거짓

System.out.println("n3 : " + n3);
```

Quiz) 삼항 연산자를 이용하여, 나이를 입력받아 성인/미성년자를 구분하는 코드를 작성해보자.



Operator

- 문제 1) 한 글자를 입력받아 글자가 산술 연산자 (+, -, *, /, %)인지 아닌지 판별하는 코드 작성

- 문제 2) 두 정수와 부호를 입력받아 답을 출력하는 코드 작성



Operator

- .(점) : 참조 연산자 - 클래스 혹은 객체의 내부 값을 참조하기 위해 사용**
- [] : 배열 참조 - 배열의 순번을 표현할 때 사용**
- () : 괄호 - 특정 수식을 묶어 먼저 처리할 때 사용**
- ! : 논리 부정 - 논리 자료형의 데이터의 앞에서 결과를 반전**
- ~ : 비트 부정 - 정수 데이터를 비트단위로 반전**
- <<, >> : 비트 Shift 연산 - 정수를 비트로 변환하여 자리수 이동**