

JAVA 연산자

프로그램은 사람이 이해하는 코드를 작성.
느려도 꾸준하면 경기에서 이긴다.

Content

3. 연산자

1. 연산자
2. 산술연산자
3. 증감연산자

“식을 구성하는 기본 단위”

연산자 (Operator)

- 연산 결과: 산술, 증감, 비트,시프트 연산자
- 연산 결과가 참/거짓: 비교, 논리 연산자
- 연산 결과가 아닌 대입: 대입, 삼항 연산자

연산자 (Operator)	연산기호	기능	결과
산술 연산자(arithmetic operator)	+, -, *, /, %	사칙연산, 나머지 연산 모두 두 개의 피연산자를 가지는 이항 연산자 결합 방향은 왼쪽에서 오른쪽	값
증감 연산자(increment and decrement operators)	++, --	값이 1 씩 증가 및 감소	값
비트 연산자(bitwise operator)	&, , ~, ^	비트 AND, OR, NOT, XOR	값
시프트 연산자(shift operator)	>>, <<, >>>	비트 단위로 이동	값
비교 연산자(comparison operator)	<, >, <=, >=, ==, !=	값의 비교	참/거짓
논리 연산자(logical operator)	&&, , !, ^	논리 AND, OR, NOT, XOR	참/거짓
대입 연산자(assignment operator)	=, +=, -=, *=, /=, &=, =, >>=, <<=, >>>=	산술연산 결과를 대입 (축약)	실행
삼항 연산자(ternary operator)	(조건) ? 참실행 : 거짓실행	조건이 참이면 참실행, 거짓이면 거짓 실행	실행
instanceof 연산자	instanceof	객체가 어떤 클래스인지, 어떤 클래스를 상속받았는지 확인하는데 사용하는 연산자	참/거짓

“식을 구성하는 기본 단위”

연산자 우선 순위

- 연산자의 우선순위는 수식 내에 여러 연산자가 함께 등장할 때, 어느 연산자가 먼저 처리될 것인가를 결정

위	연산자	설명	결합 방향
1	[]	첨자 연산자	왼쪽에서 오른쪽으로
	.	멤버 연산자	왼쪽에서 오른쪽으로
2	++	후위 증가 연산자	왼쪽에서 오른쪽으로
	--	후위 감소 연산자	왼쪽에서 오른쪽으로
3	!	논리 NOT 연산자	오른쪽에서 왼쪽으로
	~	비트 NOT 연산자	오른쪽에서 왼쪽으로
	+	양의 부호 (단항 연산자)	오른쪽에서 왼쪽으로
	-	음의 부호 (단항 연산자)	오른쪽에서 왼쪽으로
	++	전위 증가 연산자	오른쪽에서 왼쪽으로
	--	전위 감소 연산자	오른쪽에서 왼쪽으로
	(타입)	타입 캐스트 연산자	오른쪽에서 왼쪽으로
4	*	곱셈 연산자	왼쪽에서 오른쪽으로
	/	나눗셈 연산자	왼쪽에서 오른쪽으로
	%	나머지 연산자	왼쪽에서 오른쪽으로
5	+	덧셈 연산자 (이항 연산자)	왼쪽에서 오른쪽으로
	-	뺄셈 연산자 (이항 연산자)	왼쪽에서 오른쪽으로
6	<<	비트 왼쪽 시프트 연산자	왼쪽에서 오른쪽으로

위	연산자	설명	결합 방향
6	>>	부호 비트를 확장하면서 비트 오른쪽 시프트	왼쪽에서 오른쪽으로
	>>>	부호 비트까지 모두 비트 오른쪽 시프트	왼쪽에서 오른쪽으로
7	<	관계 연산자(보다 작은)	왼쪽에서 오른쪽으로
	<=	관계 연산자(보다 작거나 같은)	왼쪽에서 오른쪽으로
	>	관계 연산자(보다 큰)	왼쪽에서 오른쪽으로
	>=	관계 연산자(보다 크거나 같은)	왼쪽에서 오른쪽으로
	instanceof	인스턴스의 실제 타입 반환	왼쪽에서 오른쪽으로
8	==	관계 연산자(와 같은)	왼쪽에서 오른쪽으로
	!=	관계 연산자(와 같지 않은)	왼쪽에서 오른쪽으로
9	&	비트 AND 연산자	왼쪽에서 오른쪽으로
10	^	비트 XOR 연산자	왼쪽에서 오른쪽으로
11		비트 OR 연산자	왼쪽에서 오른쪽으로
12	&&	논리 AND 연산자	왼쪽에서 오른쪽으로
13		논리 OR 연산자	왼쪽에서 오른쪽으로
14	?:	삼항 조건 연산자	오른쪽에서 왼쪽으로
15	=	대입 연산자 및 복합 대입 연산자 (=, +=, -=, *=, /=, %=, <<=, >>=, >>>=, &=, ^=, =)	오른쪽에서 왼쪽으로

“ 기본 적인 연산자 ”

산술 연산자

- 사칙 연산 (+, -, *, /) 와 나머지 연산 (%)
- 나누기 연산(/)은 몫에 대한 결과, 나머지를 구하기 위해서는 나머지 연산 (%)를 사용 함 (모듈로 연산)

```
public class ArithmeticOperator {  
    public static void main(String[] args) {  
        int numA = 10;  
        int numB = 3;  
  
        System.out.println("numA + numB = " + (numA + numB) );  
        System.out.println("numA - numB = " + (numA - numB) );  
        System.out.println("numA * numB = " + (numA * numB) );  
        System.out.println("numA / numB = " + (numA / numB) );  
        System.out.println("numA % numB = " + (numA % numB) );  
  
        double doubleA = 10.27;  
        double doubleB = 3.5;  
  
        System.out.println("doubleA + doubleB = " + (doubleA + doubleB) );  
        System.out.println("doubleA - doubleB = " + (doubleA - doubleB) );  
        System.out.println("doubleA * doubleB = " + (doubleA * doubleB) );  
        System.out.println("doubleA / doubleB = " + (doubleA / doubleB) );  
        System.out.println("doubleA % doubleB = " + (doubleA % doubleB) );  
    }  
}
```

```
numA + numB = 13  
numA - numB = 7  
numA * numB = 30  
numA / numB = 3  
numA % numB = 1  
doubleA + doubleB = 13.77  
doubleA - doubleB = 6.77  
doubleA * doubleB = 35.945  
doubleA / doubleB = 2.934285714285714  
doubleA % doubleB = 3.2699999999999996
```