**[ 3 ] 연산자**

목표 : 자바에서 활용되는 연산자의 종류를 안다.

프로그래밍에서 각각의 연산자에 따라 처리를 효과적으로 할 수 있다.

1. 연산자(operator)란?

일정한 규칙을 가지고 계산하는 것으로 우리가 일상생활에서 사용하고 있는 덧셈, 뺄셈, 곱셈, 나눗셈을 비롯하여 프로그램에서는 자동증감, 동등비교, 비트연산 등 많은 연산자가 있다.

2. 연산자의 종류

① 산 술 연 산 : \* / + -

② 증감연산 : ++ --

③ 동등관계(비교)연산 : ==(같다) !=(다르다) < <= > >=

④ 논 리 연 산 : && || & | !

⑤ 삼항(3항)연산: ? :

⑥ 할당(대입) 연산자 : = += -= \*= /= %=

3. 우선순위에 따른 연산자 종류

① 1차 연산자 : ( )

② 단항연산자 : ! ++ --

③ 이항연산자 中 승법연산자 : \* / %

④ 이항연산자 中 가법연산자 : + -

⑤ 이항연산자 中 관계연산자 : < <= > >= == !=

⑥ 이항연산자 中 비트곱연산자 : &

⑦ 이항연산자 中 비트합연산자 : |

⑧ 이항연산자 中 논리곱연산자 : &&

⑨ 이항연산자 中 논리합연산자 : ||

⑩ 조건연산자(3항연산자) : ? :

⑪ 할당연산자 : = += -= \*= /= %=

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 순위 | 명칭 | | 연산자 | 결합성 |
| ① | 1 차 연 산 자 | | ( ) | 좌결합성 → |
| ② | 단 항 연 산 자 | | ! ++ -- | 우결합성 ← |
| ③ | 이  항  연  산  자 | 승법연산자 | \* / % | → |
| ④ | 가법연산자 | + - |
| ⑤ | 관계(비교)연산자 | < <= > >= == != |
| ⑥ | 비트곱연산자 | & |
| ⑦ | 비트합연산자 | | |
| ⑧ | 논리곱연산자 | && |
| ⑨ | 논리합연산자 | || |
| ⑩ | 조건(3항)연산자 | | ? : |
| ⑪ | 할 당 연 산 자 | | = += -= \*= /= %= | ← |

4. 예제들

(ex1) 산술연산자

**public** **class** Ex1 {

**public** **static** **void** main(String[] args) {

System.***out***.println("==== 산술 연산자 ====");

**int** n1 = 33;

**int** n2 = 10;

**int** result;

**double** resultDouble;

result = n1 + n2;

System.***out***.printf("%d %c %d = %d\n\n", n1, '+', n2, result);

result = n1 - n2;

System.***out***.printf("%d %c %d = %d\n\n", n1, '-', n2, result);

result = n1 \* n2;

System.***out***.printf("%d %c %d = %d\n\n", n1, '\*', n2, result);

result = n1 / n2;

resultDouble = (**double**)n1 / n2;

// resultDouble = (**double**)(n1 / n2);

System.***out***.printf("%d %c %d = %d(정수형)\n", n1, '/', n2, result);

System.***out***.printf("%d %c %d = %.1f(실수형)\n\n", n1, '/', n2, resultDouble);

result = n1 % n2;

System.***out***.printf("%d %c %d = %d\n\n", n1, '%', n2, result);

// 나머지 연산자의 용도

// 짝수/홀수 판별, 배수 판별 용도

// 모든 정수에 나머지 연산자 2를 적용했을 때 값이 0이면 짝수, 1이면 홀수.

// 배수를 판별할 때에는 나머지의 값이 0인지 확인합니다

**if** ( (n1%2) == 0 )

System.***out***.printf("n1은 짝수입니다\n");

**else**

System.***out***.printf("n1은 홀수입니다\n");

**if** ( (n1%5) == 0 )

System.***out***.printf("n1은 5의 배수입니다\n");

**else**

System.***out***.printf("n1은 5의 배수가 아닙니다\n");

}

}

(ex2) 증감연산자

**public** **static** **void** main(String[] args) {

**int** n1 = 10;

**int** n2 = n1++;

System.***out***.printf("1. n1 = %d, n2 = %d\n", n1, n2);

**int** n3 = 10;

System.***out***.printf("2. n3 = %d\n", n3++);

System.***out***.printf("3. n3 = %d\n", n3);

**int** n4=10;

System.***out***.printf("4. n4 = %d\n", --n4);

System.***out***.printf("5. n4 = %d\n", n4);

}

(ex3) 동등비교, 관계연산자

**public** **static** **void** main(String[] args){

**int** n1=10; **int** n2=5;

**boolean** result;

System.***out***.println("==== 동등비교, 관계연산자 ====");

// %b : boolean 타입을 출력하는 양식 (true/false)

result = n1 > n2;

System.***out***.printf("%d %c %d 는 %b\n", n1, '>', n2, result);

result = n1 < n2;

System.***out***.printf("%d %c %d 는 %b\n", n1, '<', n2, result);

result = n1 >= n2;

System.***out***.printf("%d %s %d 는 %b\n", n1, ">=", n2, result);

result = n1 <= n2;

System.***out***.printf("%d %s %d 는 %b\n", n1, "<=", n2, result);

result = n1 == n2;

System.***out***.printf("%d %s %d 는 %b\n", n1, "==", n2, result);

result = n1 != n2;

System.***out***.printf("%d %s %d 는 %b\n", n1, "!=", n2, result);

}

(ex4) 논리연산자

// 논리연산자 (AND, OR => &&, ||)

**public** **static** **void** main(String[] args) {

**int** i=1, j=10, h=10;

System.***out***.println("&&(AND) (i>j) && (++j>h) : "+((i>j) && (++j>h)));

// F && T

System.***out***.println(j); // &&연산자의 좌항이 false일 경우 우항 실행 안 함

System.***out***.println("||(OR) (i>j) || (++j>h) : "+((i<j) || (++j>h)));

// T || T

System.***out***.println(j); // ||연산자의 좌항이 true일 경우 우항 실행 안 함

}

(ex5) 삼항연산자

**public** **static** **void** main(String[] args) {

**int** h=100;

// 삼항연산자

// 조건식의 결과에 따라 값/식을 분기하는 연산자

// 형식 -> 조건식 ? 참일경우의 값/식 : 거짓일 경우의 값/식;

// 조건식 : 식의 결과가 참/거짓으로 나눠지는 식

System.***out***.println("====삼항연산자====");

String result = ((h%2)==0) ? "짝수":"홀수";

System.***out***.println("삼항연산자 결과 h는 " + result);

}

(ex6) 대입(할당)연산자

**public** **static** **void** main(String[] args) {

System.***out***.println("====할당(대입)연산자====");

**int** n1 = 10 \* 3;

// 변형된 타입의 대입연산자

// 자기 자신에 연산한 후의 결과값을 대입할 수 있는 연산자

n1 +=10; // n1 변수의 값을 10 증가 시키는 코드...

n1 \*= 3; // n1 변수의 값에 3을 곱하는 코드

System.***out***.printf("n1 = %d\n", n1);

**int** n3, n4, n5;

n3 = n4 = n5 = 10;

System.***out***.println(n3+"\n"+n4+"\n"+n5 );

}

(ex7) 비트 연산자

// 비트 연산자

// 비트를 사용해 연산하는 연산자로서,

// 정수와 정수 형태의 연산만 가능합니다.

// AND(&), OR(|), XOR(^)

**public** **static** **void** main(String[] args) {

System.***out***.println("====비트연산자====");

**int** n1 = 10;

**int** n2 = 5;

// n1 = 0 1 0 1 0

// n2 = 0 0 1 0 1

// & = 0 0 0 0 0

**int** result = n1 & n2;

System.***out***.printf("n1 & n2 =");

System.***out***.println(result);

// n1 = 0 1 0 1 0

// n2 = 0 0 1 0 1

// | = 0 1 1 1 1

result = n1 | n2;

System.***out***.printf("n1 | n2 =");

System.***out***.println(result);

// 비트연산자 XOP(^)

// 좌항과 우항의 각 비트를 비교하여

// 서로 다른 값을 가지는 자리를 1로 돌려주는 연산자

// n1 = 0 1 0 1 0

// n2 = 0 0 1 0 1

// ^ = 0 1 1 1 1

result = n1 ^ n2;

System.***out***.printf("n1 ^ n2 =");

System.***out***.println(result);

}

(연습문제1)3의 배수 숫자 판단하기 : 사용자로부터 숫자를 입력 받고, 입력 받은 숫자가 3의 배수인지 파악

**import** java.util.Scanner;

**public** **class** Q1 {

**public** **static** **void** main(String[] args) {

// 이프로그램을 실행할 때(JVM 구동될때) 클래스가 많으면 main메소드가 있는 시점에서 실행시작

// i를 3으로 나눈 나머지가 0이면 3의 배수

Scanner scanner = **new** Scanner(System.***in***);

System.***out***.print("i ? ");

**int** i = scanner.nextInt(); // 사용자로부터 입력대기

System.***out***.println((i%3)==0 ? "입력하신 i는 3의 배수입니다":"입력하신 i는 3의 배수가 아닙니다");

scanner.close();

}

}

(연습문제2) 입력한 수가 짝수인지 홀수인지 출력

(연습문제3) 두수를 입력받아

두 수가 같은지 결과를 O나 X로 출력.

첫번째 수가 더 큰지 결과를 O나 X로 출력한다.

(연습문제4) 나이를 입력받아 입력받은 나이가 65세 이상일 때, “경로우대” 출력, 아니면 “일반”출력

(연습문제5) 국어, 영어, 수학 점수를 변수에 할당하고 각 점수를 출력하고 총점, 평균(소수점2자리까지) 출력하는 프로그램을 구현 하시오

(작성한 Q2.java,Q3.java, Q4.java, Q5.java를 [yisy0703@naver.com](mailto:yisy0703@naver.com)으로 메일 발송 하시오).