

JAVA 2강 요약

강사 박주병

1. 연산자와 피연산자

변수는 데이터를 저장하는 공간이다. 이제 이런 변수를 계산 할 수 있는 연산자를 배울 차례다.
연산자는 흔히 우리가 알고 있는 사칙연산이 대표적인 예이다. 그리고 프로그래밍에서는 추가적으로 더 많은 연산자를 제공한다. 변수의 사용법은 기존의 수학과 크게 다르지 않다. 그림1을 보면 연산자는 피연산자들을 계산하도록 되어 있다. 여기서 = 는 수학에서의 같다 라는 의미와 더불어 3+5 연산을 했는 결과를 y 변수에 대입시키는 기능도 포함되어 있다.

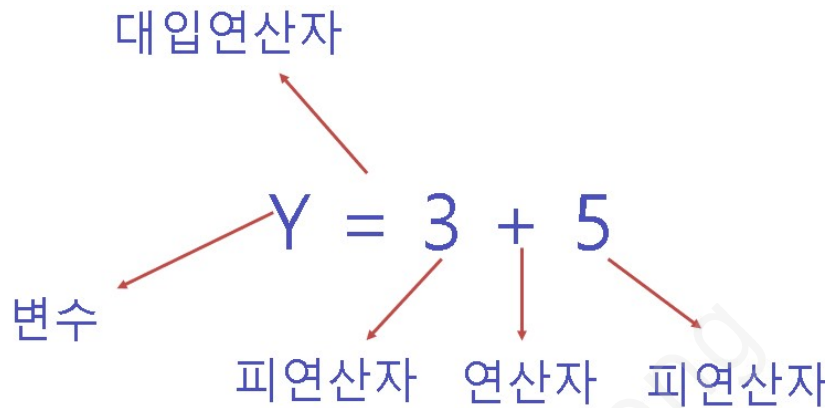


그림 1

2. 연산자의 종류

아래의 표는 프로그래밍에서 사용하는 대표적인 연산자의 종류들이다.

종류	연산자	설명
산술 연산자	+ - * / % << >>	사칙연산, 나머지 연산, 비트 연산
비교 연산자	> < >= <= == !=	크다, 작다, 같다, 다르다 등의 비교
대입 연산자	=	우변의 값을 좌변에 저장
논리 연산자	&& ! & ^ ~	조건을 연결
형변환 연산자	()	객체의 타입을 판별
삼항 연산자	?:	조건에 따라 분기
instanceof	instanceof	객체의 타입을 판별

연산자는 피연산자의 개수에 따라 분류해 볼 수 있다. 단항연산자는 피연산자가 1개, 이항연산자는 2개, 삼항연산자는 3개 이다.

단항연산자 ++3 , --7

이항연산자 3+5

삼항연산자 `System.out.println(3==5? "같다": "다르다");`

가. 산술변환

이항연산자가 수행될 때는 피연산자가 2개 필요하다. 그런데 피연산자들끼리 데이터타입이 다르면 서로 일치시킨 다음 연산을 수행해야 한다. 이럴 때 피연산자의 타입을 변환하는 것을 산술변환 이라고 한다.

그림4는 float형을 연산하기위해 산술변환을 명시적으로 하고 있다. 물론 자동 타입캐스팅이 가능한 경우라면 적어 주지 않아도 자동으로 산술변환을 한다. 아래의 경우 int 가 더 큰 형태인 float로 바뀌면 되기에 자동타입캐스팅이 될 수 있어 명시적형변환을 빼도 무관하다.

```
int a = 10;
float b = 20.0f;

float result = (float)a+b;
```

그림 4

나. 단항연산자

단항연산자는 앞에서 배웠듯이 피연산자가 1개뿐인 연산자이다.

이는 크게 피연산자의 앞에 오는 전위형과 피연산자의 뒤에 오는 후위형이 있다.

```
int a = 10;
int b = ++a;
```

전위형

```
int a = 10;
int b = a--;
```

후위형

전위형의 경우 a의 값을 먼저 1증가 시켜 11이 되고 b에다가 a의값을 넣는 것이다 따라서 a,b 둘 다 11의 값을 가진다. 후위형의 경우 a의값을 먼저 b에 넣고 a의값을 1감소 시킨다. 따라서 a는 9 b는 10의 값을 가진다.

다. 사칙연산자

사칙연산자는 기존의 수학에서 배운것과 크게 다르지 않다. 다만 곱하기의 형태가 다르며 나머지연산자가 있다.

```
int result1 = 10+3;
int result2 = 10-3;
int result3 = 10*3;
int result4 = 10/3;
int result5 = 10%3;
```

* 0으로 나눌시 에러를 발생시킨다.

```
Problems @ Javado
<terminated> main [Java
13
7
30
3
1
```

위의 그림에서 *가 곱하기 이며 %는 10을 3으로 나눈 나머지를 구하는 나머지연산자이다.

라. 비교연산자

값을 비교하여 true, false 둘 중 하나의 값을 반환하는 연산자이다. 여기서 true, false는 문자열을 의미하는 것이 아닌 boolean 타입을 의미한다. 비록 출력하면 문자열처럼 보이지만 데이터 타입은 엄밀히 1바이트 크기의 boolean 타입의 데이터 이다.

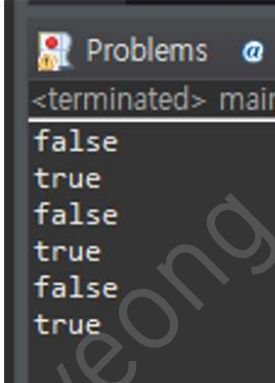
아래의 예시를 보면 수학과 크게 다를 것 없으나 == 와 !=는 눈여겨보자

우선 ==는 수학에서의 = 과 같이 “같다” 라는 의미로 쓰인다. 피연산자들의 값이 일치할 경우 true를 반환한다.

!= 는 “같이 않다” 라는 의미이다. 피연산자들의 값이 다르면 true를 반환한다.

```
int a=10;
int b=20;

boolean isEqual = a>b;
boolean isEqual2 = a<b;
boolean isEqual3 = a>=b;
boolean isEqual4 = a<=b;
boolean isEqual5 = a==b;
boolean isEqual6 = a!=b;
```



마. 논리연산자

두 개의 피연산자에 boolean 타입을 반환하는 수식이 올수 있으며 두 개의 boolean 타입을 비교하여 최종적으로 true, false를 반환한다.

논리 연산자

&&(AND) -> 피연자 모두 true 이면 true를 반환한다.

|| (OR) -> 피연자 둘중 하나만 true이면 true를 반환한다.

```
int x=15;

System.out.println(true && true);
System.out.println(true && false);
System.out.println(true || false);
System.out.println(3==1 || 10<5);
System.out.println(x>10 && x<20);
```

*논리 연산자들 끼리쓰였을 때 우선순위는 &&가 더 높다는 것을 유의하자

바. 키보드로부터 값을 받는 Scanner 클래스 사용법

앞으로 실습을 위해 불가피하게 아직 배우지 않은 클래스, 객체를 사용을 우선해야 한다. 아래의 빨간네모들이 Scanner 클래스의 사용방법이며 자세한 내용은 차후에 배우도록하며 현재는 값을 입력 받는 방법이다 라고만 생각하자.

```
package joo;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        char ch;
        Scanner scanner = new Scanner(System.in); //입력을 받기 위한 스캐너 객체를 생성

        System.out.println("값을 입력하세요:");
        String input = scanner.nextLine(); //키보드로부터 값을 입력 받아 input 변수에 저장한다.
        ch = input.charAt(0); //input 변수의 타입을 char로 변경한다.

        |

        System.out.println(ch);

    }

}
```

import java.util.Scanner; -> Scanner 클래스를 사용하기 위해 라이브러리를 가져오는 코드이다.

Scanner scanner = new Scanner(System.in); -> 스캐너 객체를 담기 위한 변수를 선언하고 객체를 생성하여 해당변수에 저장하는 것이다.

String input = scanner.nextLine(); -> String타입은 문자열을 저장하기 위한 데이터타입이다. 따라서 input은 문자열을 저장할수 있는 변수이다. scanner.nextLine() 는 키보드로부터 데이터를 입력받기위해 프로그램이 멈추는 기능을한다. 이때 콘솔창에 데이터 입력후 엔터를 쳐야지만 다음으로 프로그램 실행이 넘어간다. 그리고 입력받은 데이터를 input 변수에 저장한다.

사. 논리 부정연산자

true나 false를 !를 사용하여 반대로 값을 변환한다. !true 이면 false 가되며 !false 이면 true가 된다.

```
System.out.println(!true && false);
```



```
<terminated>
true
```

아. 조건연산자(삼항연산자)

JAVA에서 유일한 삼항연산자이며 앞에서 배웠듯이 피연산자 3개가 필요하다

조건식 ? 코드1 : 코드2 이처럼 사용하며 조건식이 true이면 코드1을 실행하며 false 이면 코드2를 실행한다.

```
char result = 10==20 ? 'a' : 'b';

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

위의 식을 보면 10과 20은 다르기에 false 가 나오며 false 이므로 'b'를 result 변수에 저장한다. 따라서 b가 출력된다.

조건 연산자는 중첩하여 사용 할 수도 있지만 너무 많은 중첩은 코드의 가독성을 떨어트리므로 그럴 경우 차후 배울 분기문인 if문을 쓰는 것이 좋다.

양수이면 1
0이면 0
음수이면 -1

```
int a=10;  
int result = a>0 ? 1 : ( a==0 ? 0 : -1);  
  
System.out.println( result);
```

자. 대입연산자

지금까지 가장 많이 사용해왔던 연산자이다. = 이며 우변에 있는 값을 좌변에 저장하는 기능이다.

```
int a = 10;
```

```
17  
18 int a = 10;  
19 System.out.println( a=30);  
20 System.out.println( a);  
21
```

- 1. 대입연산자 역시 값을 반환한다.
- 2. 대입연산자의 왼쪽은 값이 변경가능한것만 올수 있다.
(리터럴, 상수 불가능)

차. 복합대입연산자

대입연산자와 산술연산자를 결합하여 코드를 간략하게 작성 할 수도 있다. 현업에서 많이 사용한다. 아래의 그림을 보면 크게 어렵지 않을 것이다.

```
int a = 10;  
a +=20;  
a *=3;  
a %=10;
```

=

```
int a = 10;  
a =a+20;  
a =a*3;  
a =a%10;
```