

7/20(week3)

## 1. 자동 타입 변환

실수형 타입인 double에 정수값을 넣으면 자동으로 실수값으로 타입 변환  
ex)

```
double d = 5; // 실수값 0.5로 자동 타입 변환
```

```
float f = 5f; // 실수값 0.5로 자동 타입 변환
```

\* 작은 크기 타입을 큰 크기 타입으로 변환하려고 하면 자동으로 타입 변환

## 2. 강제 타입 변환

데이터 앞에 (데이터타입) 형태의 변환 연산자를 붙인다.

ex)

큰 값 → 작은 값

```
byte b = (byte)300; // 데이터 손실
```

\* 큰 값을 작은 데이터 타입으로 변환하려고 하면 데이터 손실되어 손실된 값 출력

작은 값 → 큰 값

```
double d = (double)3.14f; // float 보다 큰 double타입으로 변환할 경우 데이터 손실 없음
```

데이터 타입 크기 순서(표현할 수 있는 값의 범위)

byte < short < int < long < float < double

데이터 입력

```
package week03;
```

```
import java.util.Scanner; // java.util 패키지 안의 Scanner 클래스를 가져와서 사용
```

```
public class LMS_0720 {
```

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

```
        Scanner sc = new Scanner(System.in); // 스캐너 객체 생성
        String name = sc.nextLine(); // 라인전체 문자열을 입력받음
        int age = sc.nextInt(); // 정수를 입력받음
        String school = sc.next(); // 공백 이전의 문자열을 입력받음
        String major = sc.next();
        double score = sc.nextDouble(); // 실수를 입력받음
        sc.close();
    }
```

```
}
```

next 자료형() 형태로 자료형에 따라 다르게 받아올 수 있음

ex) nextByte(), nextLong(), nextFloat() 등

## 연산자

연산자는 대표적으로 사칙연산인 +,-,\*,/ 가 대표적임  
3\*8의 수식에서 연산자는 \*, 피연산자는 3,8이다

### 1. 산술연산자

+, -, \*, /, %

- \* +(덧셈) 연산자는 문자열 연결하는 데에도 사용
- \* %(나머지) 연산자는 정수 타입만 취함

### 2. 비교 논리

비교 연산자

피연산자를 비교해 true나 false를 반환

연산자	의미
<code>x == y</code>	x와 y가 같은가?
<code>x != y</code>	x와 y가 다른가?
<code>x &gt; y</code>	x가 y보다 큰가?
<code>x &lt; y</code>	x가 y보다 작은가?
<code>x &gt;= y</code>	x가 y 이상인가?
<code>x &lt;= y</code>	x가 y 이하인가?

\* 수학 기호에서는 “=”이 같다라고 쓰지만, 프로그래밍에서는 ‘대입하다’라는 의미 따라서 “==”라는 기호가 ‘같다’라는 의미를 가진다.

### 논리 연산자

피연산자의 조건을 결합해서 true와 false를 조사

a && b (AND)

피연산자 모두 참일때만 true

a || b (OR)

피연산자중 하나만 참이면 true

a ^ b (XOR)

피연산자 두 개의 값이 서로 다르면 true

! a(부정)

피연산자의 논리값을 반대로 변환

### 3. 대입연산자

우측 값을 좌측 변수에 저장

```
int weight = 50;
```

```
weight = weight + 50; // weight += 50;과 같음 (복합 대입 연산자)
```

### 4. 증가, 감소 연산자

변수 값을 1증가 하거나 감소 하는 연산자

++a (전위): 연산 전 1증가

--a (전위): 연산 전 1감소

a++ (후위): 연산 후 1증가

a-- (후위): 연산 후 1감소

### 5. 조건 연산자

조건식 ? 연산자1 : 연산자2

ex)

```
int a = 20;
```

```
String line = a > 10 ? "BIG" : "SMALL";
```

결과는 참 이므로 "BIG" 반환

### 6. 연산자 우선순위

사칙 연산에서는 곱셈이나 나눗셈을 덧셈이나 뺄셈보다 먼저 하듯이

연산자 사이에는 우선순위가 존재함.

대부분의 연산자는 왼쪽에서 오른쪽으로 연산하는 결합 규칙을 사용하지만,

++(전위), --(전위)와 대입 연산자는 오른쪽에서 왼쪽으로 연산 함.

### 3주차 과제

#### 1. 이름 나이 학교 학과 입출력



```
<terminated> LMS_0720 [Java Application] C:\Users\Wm2002\Desktop\Weclipse\Weclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (G
이민섭
20
부천대학교
컴퓨터소프트웨어과
이름: 이민섭
나이: 20
학교: 부천대학교
학과: 컴퓨터소프트웨어과
```

#### 2. 출력값

```
3 public class Main {
4
5     public static void main(String[] args) {
6         // TODO Auto-generated method stub
7         int a = 5;
8         int b = 6;
9
10        System.out.println("a의 값: " + a++);
11        System.out.println("b의 값: " + ++b);
12        System.out.println("a+b의 값: " + (a + b));
13
14    }
15
16 }
```

출력

5

7

13(6+7)