

Week03 자바 과제

2133031 김민지

<타입 변환>

- Byte < short < int < long < float < double

① 자동 타입 변환

: 자동으로 타입 변환이 일어나는 것

Ex) **double** d1 = 1; => 정수 1를 실수 1.0으로 자동 타입 변환

double d2 = 100; => 정수 100을 실수 100.0으로 자동 타입 변환

※ 자바 컴파일러는 큰 데이터 타입으로 변환 후 연산, 대입 수행함

② 강제 타입 변환

: 큰 크기 타입을 작은 크기 타입으로 변환하는 것

Ex) **byte** b = (**byte**)300; => (큰 -> 작은) // 정수형의 300을 byte로
형 변환하면 데이터 손실

double d = (**double**)3.14f; => (작은 -> 큰) // float의 3.14를 형
변환하면 데이터 손실 없이 저장 가능

float f = (**float**)3.14; => (큰 -> 작은) // double의 3.14를 float
로 형 변환하여 f에 3.14f 저장

<데이터 입력 방법>

: import java.util.Scanner; => 컴파일러에게 알려주기 위해 쓰임
/java.util 패키지 안 Scanner 클래스를 가져와 사용

//키보드로 입력 받는 방법

// Scanner 객체 생성

```
Scanner sc = new Scanner(System.in); // 객체 생성
int x = sc.nextInt(); // 정수 - 키보드로 데이터 입력 받음
String str = sc.next(); // 문자열
double d1 = sc.nextDouble(); // 실수
```

※ next() => 공백 이전의 문자열(String) 입력 받음

nextLine() => 라인 전체 문자열(String)을 입력 받음

nextInt() => 정수 입력 받음

nextDouble() => 실수 입력 받음

<연산자> - + - * /

① 산술 연산자 (+ - * / %)

Ex) **int** a = 10; // 10.0 => int < double 이기에 자동 타입 변환
 double b = 20.0;
 System.**out**.println(a/b);

※ 피연산자의 데이터 타입 일치시킨 후 연산 수행

% (나머지) 연산자는 정수 타입만 취함

+ (덧셈) 연산자는 문자열 연결하는 데에도 사용

② 비교 연산자 / true,false

```
Ex) int a = 10;
    int b = 20;
    System.out.println(a==b);
    System.out.println(a!=b);
    System.out.println(a<=b);
    System.out.println(a>=b);
    System.out.println(a>b);
    System.out.println(a<b);
```

※ $x = 3 \Rightarrow$ 등호 한 개는 대입으로 알아들음

④ 논리 연산자 / true,false

```
Ex) // && (AND) => 둘다 참이어야 true
    // || (OR) => 하나만 맞으면 true
    // ^ (XOR) => 두개의 값이 달라야 true
    // !a (부정) => 피연산자의 논리 값을 반대로 반환
    int a = 10;
    int b = 20;
    System.out.println(a>0 ^ b<100); // true
    System.out.println(a<0 ^ b<100); // false
    System.out.println(a<0 ^ b>100); // true
    System.out.print(!(a>100)); // true
```

⑤ 대입 연산자 / 우측에 있는 연산식의 결과값(혹은 값)을 좌측의 변수에 저장

```
Ex) int weight = 50;
    weight = weight + 50; // 50+50
    weight += 50; // 복합 대입 연산자
    weight -= 20;
```

⑥ 증가, 감소 연산자 / 변수 값(피연산자)을 1 증가하거나 감소하는 연산자

++a (전위) => 대입이 되기 전 1 증가

a++ (후위) => 대입이 된 후에 1 증가

--a (전위) => 대입이 되기 전 1 감소

a--(후위) => 대입이 된 후에 1 감소

⑦ 조건 연산자 / (? :)(=삼항 연산자)

```
Ex) //조건식? 연산자1:연산자2
    int x = 20;
    String line = x > 10 ? "Big" : "Small";
    System.out.println(line);
    //결과가 참이므로 Big 출력
```

<연산자 우선 순위>

: 연산자 사이에는 우선순위 존재

대부분의 연산자는 왼쪽-오른쪽으로 연산하는 결합 규칙을 사용,

++(전위), --(전위)와 대입 연산자는 오른쪽-왼쪽으로 연산함

<첫번째 과제>

김민지

20

부천대학교

컴퓨터소프트웨어과

이름: 김민지

나이: 20

학교: 부천대학교

학과: 컴퓨터소프트웨어과

<두번째 과제>

```
1 package week03;
2
3 import java.util.Scanner;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         int a = 5;
9         int b = 6;
10
11         System.out.println("a의 값: "+ a++);
12         System.out.println("b의 값: "+ ++b);
13         System.out.println("a+b의 값: "+ (a+b));
14
15     }
16 }
17
```

a의 값 : 5

b의 값 : 7

a + b의 값 : 13

