--타입 변환--

자동 타입 변환

: 데이터 타입을 다른 데이터 타입으로 변환하는 것

```
double d1 = 1; // 정수 1를 실수 1.0으로 자동 타입 변환
double d2 = 100; // 정수 100을 실수 100.0으로 자동 타입 변환
```

* 큰 데이터 타입으로 변환 후 연산이나 대입 수행

강제 타입 변환

: 큰 허용 범위 타입은 작은 허용 범위 타입으로 자동 타입 변환될 수 없는 것(데이터 앞에 (데이터 타입) 형태 변환 연산자 붙임)

```
float f = (float)3.14; (큰 - 작은)  // double의 3.14를 float로 형 변환하여 f에 3.14f 저장

byte b = (byte)300; (큰 - 작은) // 큰 값을 작은 데이터 값으로 변환하면 데이터 손실

double d = (double)3.14f; (작은 - 큰) // float 보다 큰 double타입으로 변환하면 데이터 손실 x
```

<크키별 타입 나열>

byte(1) < short(2) < int(4) < long(8) < float(4) < double(8)

float 가 long 보다 큰 이유

:표현할 수 있는 값의 범위가 더 크다

--데이터 입력 방법--

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

Scanner scan = new Scanner(System.in); // 객체 생성

int x = sc.nextInt(); // 클래스가 제공하는 다양한 메소드로 데이터 입력 받음

next() // 공백 이전의 문자열을 입력 받음
nextLine() // 라인 전체 문자열을 입력 받음
nextInt() // 정수를 입력 받음
nextDouble() // 실수를 입력 받음
```

연산자

:값을 연산하기 위해 사용하는 부호(+,-,*,/ 대표적)

3 + 5에서 +는 연산, 3,5는 피연산자

구분	연산자	피연산자 개수
증감 연산자	++,--	1
산술 연산자	+,-,*,/,%	2
비교 연산자	>,<,>=,<=,==,!=	2
논리 연산자	&&,||,!,^	2,1
대입 연산자	=,*=,/=,+=,^=,!=,<<=,>>=,>>>=	2
시프트 연산자	>>,<<,>>>	2
비트 연산자	&,|,^,~	2,1
조건 연산자(삼항 연산자)	?:	3

1. 산술 연산자

: 덧셈(+),뺄셈(-),곱셈(*),나눗셈(/),나머지(%)

피연산자의 데이터 타입 일치시킨 후 연산 수행
나머지(%) 연산자는 정수 타입만 취함
+(덧셈)연산자는 문자열 연결하는 데에도 사용

2. 비교 연산자
:2개의 피연산자를 비교해 결과 값으로 논리 값인 true 나 false를 반환함.

수학 기호에서는 “=”이 같다라고 쓰지만, 프로그래밍에서는 “==”이 같다 라는 의미를 가짐

논리 연산자
: 피연산자의 조건을 결합해서 true 나 false를 조사하며, 논리 타입에만 사용

a && b : 좌,우 모두 참이면 참 반환
a || b : 좌,우 하나만 참이면 참 반환
a ^ b : 좌,우 두 개의 값이 달라야 참 반환
!a : 피연산자의 논리 값을 반대로 반환

3. 대입 연산자
: 우측에 있는 연산식의 결과값을 좌측의 변수에 저장함.

count = count + 10; // count +=10;과 같음 (복합 대입 연산자)

4.증가, 감소 연산자
:변수 값을 1 증가하거나 감소하는 연산자

++ a(전위): 연산 전 a 값 1 증가

-- a(전위): 연산 전 a 값 1 감소 // 증가된 값을 그 자리에서 반환하여 확인

a ++(후위): 연산 후 a 값 증가

a --(후위): 연산 후 a 값 감소 // 증가되기 전 값이 반환되어 값을 즉시 확인 불가

5. 조건 연산자
: 조건식이 true이면 결과 값은 연산식 1의 값이 되고,false이면 연산자 2의 값이 ehoa
(조건식 ? 연산자 1 : 연산자 2)

ex)

int a = 30;

String line = a >20? “True” : “False”;

결과는 참이므로 True 출력

6. 연산자 우선 순위
:연산자 사이에는 우선순위 존재

대부분의 연산자는 왼쪽에서 오른쪽으로 연산하는 결합 규칙을 사용하지만
++(전위), --(전위)와 대입 연산자는 오른쪽에서 왼쪽으로 연산

3주차 과제

첫 번째 과제

두 번째 과제